

# Virtual Chassis User Guide for EX2200, EX3300, EX4200, EX4500 and EX4550 Switches

Published  
2022-03-10

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

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

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

*Virtual Chassis User Guide for EX2200, EX3300, EX4200, EX4500 and EX4550 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 | xiii**

1

## **Overview**

**Virtual Chassis Overview | 2**

Virtual Chassis Overview for Switches | 2

Understanding EX Series Virtual Chassis | 9

Understanding Virtual Chassis Components | 21

Understanding Mixed EX Series and QFX Series Virtual Chassis | 39

Understanding How the Primary in a Virtual Chassis Is Elected | 47

Understanding Software Upgrades in a Virtual Chassis | 49

Understanding Global Management of a Virtual Chassis | 51

Understanding the High-Speed Interconnection of the Dedicated Virtual Chassis Ports Connecting EX4200, EX4500, and EX4550 Member Switches | 54

Understanding Virtual Chassis Port Link Aggregation | 55

Understanding Fast Failover in an EX Series Virtual Chassis | 57

Understanding Split and Merge in a Virtual Chassis | 65

Understanding Automatic Software Update on Virtual Chassis Member Switches | 68

Understanding Nonstop Software Upgrade on EX Series Switches | 70

Understanding Virtual Chassis Device Reachability Testing | 79

Understanding MAC Address Assignment on a Virtual Chassis | 79

Understanding High Availability on an EX Series Virtual Chassis | 80

2

## **Configuration**

**Configuration Examples | 84**

Example: Configuring an EX2200 Virtual Chassis with Four Member Switches | 85

Requirements | 85

Overview and Topology | 86

Configuration	87
Verification	89
Troubleshooting the Virtual Chassis	90

#### Example: Configuring an EX3300 Virtual Chassis with a Primary and Backup | 91

Requirements	92
Overview and Topology	92
Configuration	93
Verification	94
Troubleshooting the Virtual Chassis	96

#### Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98

Requirements	98
Overview and Topology	99
Configuration	100
Verification	101
Troubleshooting the Virtual Chassis	104

#### Example: Configuring an EX4500 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 105

Requirements	105
Overview and Topology	106
Configuration	107
Verification	108
Troubleshooting the Virtual Chassis	111

#### Example: Expanding an EX3300 Virtual Chassis | 112

Requirements	112
Overview and Topology	113
Configuration	114
Verification	116
Troubleshooting	118

#### Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet | 119

Requirements	120
Overview and Topology	120
Configuration	122

- Verification | **124**
- Troubleshooting | **126**

#### Example: Adding EX4500 Switches to a Preprovisioned Virtual Chassis | **128**

- Requirements | **128**
- Overview and Topology | **129**
- Configuration | **130**
- Verification | **132**

#### Example: Adding an EX4500 Switch to a Nonprovisioned Virtual Chassis | **134**

- Requirements | **134**
- Overview and Topology | **135**
- Configuration | **137**
- Verification | **139**

#### Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration | **140**

- Requirements | **141**
- Overview and Topology | **141**
- Configuration | **142**
- Verification | **143**
- Troubleshooting | **146**

#### Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets | **148**

- Requirements | **149**
- Overview and Topology | **149**
- Configuration | **152**
- Verification | **155**
- Troubleshooting | **158**

#### Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets | **159**

- Requirements | **159**
- Overview and Topology | **160**
- Configuration | **162**
- Verification | **166**
- Troubleshooting | **167**

#### Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File | **168**

- Requirements | **169**

Overview and Topology	170
Configuration	175
Verification	178
Troubleshooting	182

#### Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis | 183

Requirements	184
Overview and Topology	184
Configuration	186
Verification	188
Troubleshooting	190

#### Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch | 191

Requirements	191
Overview and Topology	192
Configuration	194
Verification	198
Troubleshooting	199

#### Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch | 200

Requirements	201
Overview and Topology	201
Configuring LACP for the LAGs on the Virtual Chassis Access Switch	202
Configuring LACP for the LAGs on the Virtual Chassis Distribution Switch	203
Verification	205
Troubleshooting	207

#### Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails | 208

Requirements	208
Overview and Topology	209
Configuration	211
Verification	212

#### Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge | 213

Requirements	214
--------------	-----

Overview and Topology	214
Configuration	215
Verification	216

#### Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports | 217

Requirements	217
Overview and Topology	218
Configuration	220
Verification	224
Troubleshooting	228

#### Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches | 229

Requirements	229
Overview and Topology	230
Configuration	231
Verification	232

### Configuration Tasks | 234

#### Configuring an EX2200 Virtual Chassis (CLI Procedure) | 235

Configuring an EX2200 Virtual Chassis with a Preprovisioned Configuration File	236
Configuring an EX2200 Virtual Chassis with a Nonprovisioned Configuration File	238

#### Configuring an EX3300 Virtual Chassis (CLI Procedure) | 240

Configuring an EX3300 Virtual Chassis with a Preprovisioned Configuration File	241
Configuring an EX3300 Virtual Chassis with a Nonprovisioned Configuration File	243

#### Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) | 245

Configuring an EX4200, EX4500, or EX4550 Virtual Chassis with a Preprovisioned Configuration File	246
Configuring an EX4200, an EX4500, or EX4550 Virtual Chassis with a Nonprovisioned Configuration File	248

#### Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure) | 250

Configuring an EX2200, EX2200-C, EX3300, EX4200, EX4300, EX4500, EX4550, or EX4600 Virtual Chassis (J-Web Procedure)	250
Enabling Virtual Chassis Mode on an EX8200 Switch (J-Web Procedure)	252
Configuring an EX8200 Virtual Chassis (J-Web Procedure)	252
Preprovision the Virtual Chassis	253
Configure Virtual Chassis Members	253

Configure Virtual Chassis Ports | 254

Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure) | 255

Installing Software on a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Switches (CLI Procedure) | 259

Adding a New Switch to an Existing EX3300 Virtual Chassis (CLI Procedure) | 261

Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure) | 263

Adding a New Switch to an Existing Virtual Chassis Within the Same Wiring Closet | 263

Adding a New Switch from a Different Wiring Closet to an Existing Virtual Chassis | 264

Adding a New Switch to an Existing Preprovisioned Virtual Chassis Using Autoprovisioning and Automatic VCP Conversion | 266

Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis (CLI Procedure) | 269

Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis (CLI Procedure) | 271

Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis (CLI Procedure) | 273

Removing or Replacing a Member Switch of a Virtual Chassis Configuration | 275

Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch | 277

Remove, Repair, and Reinstall the Same Switch | 278

Remove a Member Switch, Replace It with a Different Switch, and Reapply the Old Configuration | 279

Replace a Member Switch With a Different Type of Switch That Changes the Virtual Chassis to Mixed Mode | 282

Removing an EX4200, EX4500, or EX4550 Switch From a Mixed Virtual Chassis (CLI Procedure) | 283

Configuring Primary Role of a Virtual Chassis | 284

Configuring Primary Role Using a Preprovisioned Configuration File | 285

Configuring Primary Role Using a Configuration File That Is Not Preprovisioned | 286

Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port | 287

When to Configure VCPs | 288

Prepare Virtual Chassis Member Switches Before Converting VCPs | 289

Set Uplink Ports to Form VCP Links Between Member Switches in a Virtual Chassis | 290

Set an Uplink Port as a VCP on a Standalone Switch | 291



Remove a VCP Setting on an Uplink or Network Port	292
Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch (CLI Procedure)	293
Setting an Uplink Port as a Virtual Chassis Port Using the LCD Panel	294
Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis (CLI Procedure)	295
Configuring the Timer for the Backup Member to Start Using Its Own MAC Address as Primary of a Virtual Chassis	296
Configuring Fast Failover in an EX Series Virtual Chassis	297
Disabling Fast Failover in an EX Series Virtual Chassis	298
Disabling Split and Merge in a Virtual Chassis	299
Configuring Automatic Software Update on Virtual Chassis Member Switches	301
Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis (CLI Procedure)	303
Preparing the Switch for Software Installation	304
Upgrading the Software Using NSSU	305
Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge	308
Configuring Graceful Routing Engine Switchover in a Virtual Chassis	308
Resetting the EX4500 Switch to Run as a Standalone Switch (CLI Procedure)	309
Enabling VLAN Pruning for Broadcast, Multicast, and Unknown Unicast Traffic in an EX Series Virtual Chassis (CLI Procedure)	311
<b>Configuration Statements</b>	<b>312</b>
auto-conversion	313
auto-sw-update	315
fast-failover	319
graceful-restart (Enabling Globally)	320
graceful-switchover	323
id	324
location (Virtual Chassis)	325

mac-persistence-timer | 327

mastership-priority | 329

member | 331

no-management-vlan | 333

no-split-detection | 334

preprovisioned | 336

redundancy (Graceful Switchover) | 337

role | 339

serial-number | 342

traceoptions (Virtual Chassis) | 343

vc-port | 347

vcp-snmp-statistics | 348

virtual-chassis | 350

## 3

## Administration

### Routine Monitoring | 354

Command Forwarding Usage with EX Series and QFX Series Virtual Chassis | 354

Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member | 362

Verifying That Virtual Chassis Ports Are Operational | 364

Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366

Verifying That Graceful Routing Engine Switchover Is Working in the Virtual Chassis | 369

Verifying the Setting for the PIC Mode on an EX4500 Switch in a Virtual Chassis | 371

Verifying the Setting for the Virtual Chassis Mode on EX4200, EX4500, and EX4550 Switches | 372

Verifying Connectivity Between Virtual Chassis Member Devices | 374

### Operational Commands | 376

clear virtual-chassis vc-port statistics | 377

request chassis pic-mode | 379

request session member | 382

request system software nonstop-upgrade | 383

request virtual-chassis device-reachability | 396

request virtual-chassis mode | 399

request virtual-chassis recycle | 405

request virtual-chassis renumber | 407

request virtual-chassis vc-port (Dedicated VCP) | 408

request virtual-chassis vc-port | 411

show chassis nonstop-upgrade | 414

show chassis pic-mode | 417

show system uptime | 419

show virtual-chassis active-topology | 426

show virtual-chassis device-topology | 433

show virtual-chassis protocol adjacency | 441

show virtual-chassis protocol database | 446

show virtual-chassis protocol interface | 452

show virtual-chassis protocol route | 455

show virtual-chassis protocol statistics | 459

show virtual-chassis fast-failover | 464

show virtual-chassis login | 466

show virtual-chassis mode | 467

show virtual-chassis | 471

show virtual-chassis vc-path | 480

show virtual-chassis vc-port | 485

show virtual-chassis vc-port statistics | 492

## Troubleshooting Procedures | 504

### Troubleshooting an EX Series Virtual Chassis | 504

A Disconnected Member Switch's ID Is Not Available for Reassignment | 504

Load Factory Default Does Not Commit on a Multimember Virtual Chassis | 505

The Member ID Persists When a Member Switch Is Disconnected From a Virtual Chassis | 506

A Member Switch Is Not Participating in a Mixed Virtual Chassis | 507

Unknown Traffic Looping Occurs After Configuring an Uplink Port as a Redundant VCP with a Dedicated VCP | 509

# About This Guide

Use this guide to set up and configure an EX2200, EX3300, EX4200, EX4500, or EX4550 Virtual Chassis. A Virtual Chassis is composed of a supported combination of multiple switches that operate and are managed as a single switch or network entity. Refer also to the hardware documentation for the types of switches comprising the Virtual Chassis for more details on how to physically interconnect them.

# 1

PART

## Overview

---

[Virtual Chassis Overview](#) | 2

---

# Virtual Chassis Overview

## IN THIS CHAPTER

- Virtual Chassis Overview for Switches | 2
- Understanding EX Series Virtual Chassis | 9
- Understanding Virtual Chassis Components | 21
- Understanding Mixed EX Series and QFX Series Virtual Chassis | 39
- Understanding How the Primary in a Virtual Chassis Is Elected | 47
- Understanding Software Upgrades in a Virtual Chassis | 49
- Understanding Global Management of a Virtual Chassis | 51
- Understanding the High-Speed Interconnection of the Dedicated Virtual Chassis Ports Connecting EX4200, EX4500, and EX4550 Member Switches | 54
- Understanding Virtual Chassis Port Link Aggregation | 55
- Understanding Fast Failover in an EX Series Virtual Chassis | 57
- Understanding Split and Merge in a Virtual Chassis | 65
- Understanding Automatic Software Update on Virtual Chassis Member Switches | 68
- Understanding Nonstop Software Upgrade on EX Series Switches | 70
- Understanding Virtual Chassis Device Reachability Testing | 79
- Understanding MAC Address Assignment on a Virtual Chassis | 79
- Understanding High Availability on an EX Series Virtual Chassis | 80

## Virtual Chassis Overview for Switches

### IN THIS SECTION

- Benefits of Virtual Chassis on Switches | 4
- Virtual Chassis Basics on Switches | 4

- [Global Management of Member Switches in a Virtual Chassis | 7](#)
- [High Availability Using Redundancy | 7](#)
- [Adaptability as an Access Switch or Distribution Switch | 8](#)
- [Virtual Chassis Provisioning From the Factory-Default State Using the Phone-Home Client | 8](#)

Many Juniper Networks EX Series and QFX Series switches support *Virtual Chassis*, a flexible and scalable technology with which you can connect individual switches together to form one unit, and configure and manage the unit as a single chassis. Virtual Chassis ports (VCPs) connect member switches together to form a Virtual Chassis, and are responsible for passing all data and control traffic between member switches.

Use the following links to find the right Virtual Chassis User Guide for different EX Series and QFX Series switches if this guide doesn't cover the switches you're interested in:

- [Virtual Chassis User Guide for EX2200, EX3300, EX4200, EX4500 and EX4550 Switches](#) covers configuring and maintaining EX2200, EX3300, EX4200, EX4500, and EX4550 Virtual Chassis.
- [Virtual Chassis User Guide for EX8200 Switches](#) describes configuring and maintaining EX8200 Virtual Chassis.
- For deployments with EX9200 switches, you should plan or move to MC-LAG or Junos Fusion Enterprise architectures rather than using a Virtual Chassis. We do not recommend using EX9200 switches in a Virtual Chassis. If needed to aid in migration away from EX9200 Virtual Chassis, see [Virtual Chassis User Guide for EX9200 Switches](#).
- [Virtual Chassis User Guide for Switches](#) describes configuring and maintaining *all other* EX Series and QFX Series Virtual Chassis.

**NOTE:** Virtual Chassis Fabric (VCF) is an evolution of Virtual Chassis technology. VCF configurations share some elements of Virtual Chassis technology, including configuration statements and administrative commands, but use a spine-and-leaf topology with supported QFX Series switches as spine devices and supported EX Series and QFX Series switches as leaf devices. For details on configuring and maintaining a VCF, see the [Virtual Chassis Fabric User Guide](#).



## Benefits of Virtual Chassis on Switches

- Simplifies configuration and maintenance: Multiple devices can be managed as a single device with the same or similar capabilities as the standalone device.
- Increases fault tolerance and high availability (HA): A Virtual Chassis can remain active and network traffic can be redirected to other member switches when a single member switch fails.
- Flattens your network and reduces networking overhead by allowing network devices to synchronize to one resilient logical device rather than to multiple physical devices.
- Enables a simplified Layer 2 network topology that minimizes or eliminates the need for loop prevention protocols such as Spanning Tree Protocol (STP).
- Provides a flexible model for expanding your network: You can easily add Virtual Chassis member switches to increase the number of access ports on your network to support more servers, computers, phones, or other devices with minimal complications to the existing network topology and switch configuration.

## Virtual Chassis Basics on Switches

Virtual Chassis is a feature on Juniper Networks routing or switching devices that provides network resiliency in the form of redundant Routing Engines and network expansion flexibility with minimal impact to a configuration.

Virtual Chassis technology on switches enables you to interconnect supported combinations of EX Series and QFX Series switches into one logical device that you can configure and manage as a single unit. Switches interconnected into a Virtual Chassis are called Virtual Chassis member switches, identified by a member ID within the Virtual Chassis.

Virtual Chassis member switches are interconnected and communicate with each other using Virtual Chassis ports (VCPs).

## Connecting Member Switches with Virtual Chassis Ports

A switch is not recognized by the Virtual Chassis as a member switch until it is interconnected with the primary or interconnected with an existing member of the Virtual Chassis using VCPs. EX Series and QFX Series switches that can be in a Virtual Chassis might support one or more of the following VCP options:

- Network or uplink ports that support the option to be configured as VCPs. Most switches support this VCP option.
- Ports that are configured as VCPs in the default factory settings. These ports can also be converted into and used as network ports instead of as VCPs, and converted back into VCPs again if needed.

- Dedicated VCPs, which are ports that can only function as VCPs. Only a few switches have dedicated VCPs.

Available VCP options vary among the different switch models. See *Virtual Chassis Port Options* for a summary of the ports that are supported as VCPs on different switches.

When a port is set as a VCP, it cannot be used for any other purpose. If you want to use the port for another purpose, you must delete the VCP setting using the `request virtual-chassis vc-port` command. You can run this command directly on the member whose uplink VCP setting you want to delete or through the primary member of the Virtual Chassis configuration.



**CAUTION:** Deleting a VCP in a Virtual Chassis configuration can cause the Virtual Chassis configuration to split. For more information, see *Understanding Split and Merge in a Virtual Chassis*.

If redundant VCP links of the same speed are connected between the same two member switches of a Virtual Chassis, the ports automatically form a VCP Link Aggregation Group (LAG) or bundle that distributes the inter-member VCP traffic load among them. See *Understanding Virtual Chassis Port Link Aggregation* for details.

## Virtual Chassis Configuration

You configure and manage nearly all aspects of an EX Series or QFX Series Virtual Chassis through the primary switch of the Virtual Chassis. However, you can also configure Virtual Chassis parameters when a switch is a standalone switch not interconnected with other members yet, because any switch that supports being in a Virtual Chassis is by default a single-member Virtual Chassis with member ID 0. Upon connecting the switch with others in a Virtual Chassis, any Virtual Chassis configuration statements and uplink Virtual Chassis port (VCP) settings previously configured on the standalone switch remain part of its configuration.

You can set up an EX Series or QFX Series Virtual Chassis using a nonprovisioned or preprovisioned configuration. If you want to deterministically control the roles and member IDs assigned to the member switches when creating and managing a Virtual Chassis, use a preprovisioned configuration, which distinguishes member switches by associating their serial numbers with the member ID.

When adding new member switches to a preprovisioned Virtual Chassis, you might be able to simplify the procedure by using the autoprovisioning feature, which automatically converts the interconnecting links into VCPs when cabling the new switch into the Virtual Chassis under certain conditions and configuration settings. See *Automatic Virtual Chassis Port (VCP) Conversion* for details.

## Configuring Interfaces for a Virtual Chassis

The member ID of an EX Series or QFX Series Virtual Chassis member switch functions as an FPC slot number. When you are configuring interfaces for a Virtual Chassis configuration, you specify the appropriate member ID as the *slot* element of the interface name.

The default factory settings for a Virtual Chassis configuration include FPC 0 as a member of the default VLAN because FPC 0 is configured as part of the ethernet-switching family. To include the FPC in the default VLAN, add the ethernet-switching family to the configurations for those interfaces.

## Mixed and Non-mixed EX Series and QFX Series Virtual Chassis

A Virtual Chassis might consist of all the same type of switches or different types of switches in supported combinations.

Some combinations of switches in a Virtual Chassis comprise a *mixed Virtual Chassis*, which contains member switches that have operational differences requiring the Virtual Chassis to be configured with a mixed mode setting that enables all of the member switches to inter-operate successfully.

Some combinations of different types or models of switches can inter-operate in a Virtual Chassis without requiring the Virtual Chassis to be configured in mixed mode, such as different switches that can run the same Junos OS software image.

See *Understanding Mixed EX Series and QFX Series Virtual Chassis* for details on the different combinations of switches supported in a Virtual Chassis.

## Virtual Chassis Member Switch Roles

Member switches in an EX Series or QFX Series Virtual Chassis operate in either a *primary* Routing Engine role, *backup* Routing Engine role, or *linecard* role. For some mixed Virtual Chassis, the member switches in the Routing Engine role are recommended (and in some cases required) to be particular types or models of switches. Any switch supported in a Virtual Chassis can operate in the linecard role.

A standalone switch that supports Virtual Chassis is by default a single-member Virtual Chassis that is assigned member ID 0 and operates in the primary Routing Engine role as the primary of itself. When connected and configured into a Virtual Chassis with other member switches, the switch will be assigned a unique member ID and might take on a different role.

A nonprovisioned Virtual Chassis uses a primary-role election algorithm to select the member switches that assume the primary and backup roles if an existing member switch in the Routing Engine role fails or when new member switches are added. In a preprovisioned Virtual Chassis, you assign the roles to each member switch when forming the Virtual Chassis and adding or replacing member switches.

See the following for details on EX Series and QFX Series Virtual Chassis member switch roles:

- *Understanding Virtual Chassis Components*
- *Understanding How the Primary in a Virtual Chassis Is Elected*

## Global Management of Member Switches in a Virtual Chassis

The interconnected member switches in a Virtual Chassis operate and can be configured as a single network entity.

The serial console port and dedicated out-of-band management port on individual switches have global virtual counterparts when the switches are interconnected in a Virtual Chassis configuration. You can connect to the primary switch by connecting a terminal directly to the console port of any member switch. A *virtual management Ethernet (VME)* interface allows you to remotely manage the Virtual Chassis configuration by connecting to the out-of-band management port of any member switch through a single IP address. You can perform remote configuration and administration of all member switches of the Virtual Chassis configuration using the Junos CLI through the VME interface. See *Understanding Global Management of a Virtual Chassis* for details.

When setting up a Virtual Chassis on EX Series switches that support the EZSetup script, you can run EZSetup once to specify the identification parameters for the primary, and these parameters implicitly apply to all member switches of the Virtual Chassis.

On switches that support the J-Web user interface, you can view the Virtual Chassis as a single device in J-Web and apply various device management functions to all member switches of the Virtual Chassis.

## High Availability Using Redundancy

Interconnecting EX Series or QFX Series switches into a Virtual Chassis increases your network's high availability. A Virtual Chassis is more fault tolerant than a standalone switch because it can remain active, forward traffic and provide sub-second convergence in the case of a device or link failure.

Standalone switches that support only a single Routing Engine can operate with a primary and a backup Routing Engine when configured into a Virtual Chassis, and therefore support some high availability features that would otherwise not be available on the switch, such as Graceful Routing Engine Switchover (GRES) for hitless failover.

You can also increase fault tolerance within a Virtual Chassis by configuring other supported high availability features. For example, configuring Link Aggregation Group (LAG) bundles that include member links on different switches in the same Virtual Chassis enables traffic traversing the LAG to be redirected from a Virtual Chassis member switch that fails to links on another active Virtual Chassis member switch.

## Adaptability as an Access Switch or Distribution Switch

A Virtual Chassis configuration supports a variety of user environments because it can be composed of different types of switches. You can select different switch models to support various functions. For example, you might set up one Virtual Chassis access switch configuration composed of full Power over Ethernet (PoE) models to support users sitting in cubicles equipped with PCs and Voice over IP (VoIP) phones. You could set up another Virtual Chassis configuration with partial PoE models to support the company's internal servers, and another to support the company's external servers. You can alternatively use a Virtual Chassis in a topology as a distribution switch.

## Virtual Chassis Provisioning From the Factory-Default State Using the Phone-Home Client

Phone-home provisioning on a Virtual Chassis is a form of zero-touch provisioning (ZTP). With phone-home provisioning, when a device in the factory-default state boots up, a phone-home client (PHC) process automatically starts running on the device. The PHC gets bootstrapping information over the network from a central network management data source called the phone-home server (PHS), and installs the intended software image and configuration on the device without requiring any user interaction at the remote site.

The PHC also supports phone-home provisioning on some EX Series Virtual Chassis. Check [Feature Explorer](#) and search for **phone-home** to see the Virtual Chassis platforms that support phone-home provisioning. Phone-home provisioning on a Virtual Chassis is an extension of standalone device phone-home support.

See [Provision a Virtual Chassis Using the Phone-Home Client](#) for details about how the PHC works to provision a Virtual Chassis.

The PHS is usually part of a network management system (NMS) that supports phone-home provisioning. The network administrator enters the information that defines how the devices or Virtual Chassis at remote sites should be set up. An organization might have more than one PHS for redundancy.

## Requirements for Phone-Home Provisioning to Work for a Virtual Chassis

The PHC only supports Virtual Chassis that meet the following conditions:

- The Virtual Chassis member devices have dedicated or default-configured VCPs.

*Virtual Chassis Port Options* explains the different VCP types and what's supported on different platforms.

- The Virtual Chassis members are all the same type of device.

In other words, it isn't a mixed-mode Virtual Chassis. See *Understanding Mixed EX Series and QFX Series Virtual Chassis*.

- All member devices have the factory-default configuration.

The PHC process only runs when a device or Virtual Chassis is in the factory-default state.

- The Virtual Chassis member are interconnected in a ring topology using only dedicated or default-configured VCPs.

[How To Enable Phone-Home Provisioning on a Virtual Chassis](#) shows a sample EX4300 Virtual Chassis that is wired this way.

- At least one Virtual Chassis member has an active connection to the network and can access an available PHS that supports provisioning a Virtual Chassis.

The connection to the PHS can be through the Virtual Chassis VME interface or any network-facing port on any Virtual Chassis member. See *Understanding Global Management of a Virtual Chassis* for more about how the VME interface works.

## RELATED DOCUMENTATION

*Understanding EX Series Virtual Chassis*

*Understanding QFX Series Virtual Chassis*

*Understanding Virtual Chassis Components*

*Understanding Global Management of a Virtual Chassis*

*Understanding How the Primary in a Virtual Chassis Is Elected*

*Understanding High Availability on an EX Series Virtual Chassis*

## Understanding EX Series Virtual Chassis

### IN THIS SECTION

- [Virtual Chassis Support on EX Series Switches](#) | 10
- [Basic Configuration of EX Series Virtual Chassis](#) | 13
- [EX2200 Switches in a Virtual Chassis](#) | 14
- [EX2300 Switches in a Virtual Chassis](#) | 15

- [EX3300 Switches in a Virtual Chassis | 15](#)
- [EX3400 Switches in a Virtual Chassis | 16](#)
- [EX4200, EX4500, and EX4550 Switches in a Virtual Chassis | 16](#)
- [EX4300 Switches in a Virtual Chassis | 17](#)
- [EX4400 Switches in a Virtual Chassis | 18](#)
- [EX4600 Switches in a Virtual Chassis | 19](#)
- [EX4650 Switches in a Virtual Chassis | 20](#)

This topic introduces EX Series Virtual Chassis. An EX Series Virtual Chassis is a supported combination of interconnected EX2200, EX2300, EX3300, EX3400, EX4200, EX4500, EX4550, EX4300, EX4600, EX4400, or EX4650 switches operating as one logical device and managed as a single chassis. Switches in a Virtual Chassis are called *member switches*.

For information on EX8200 Virtual Chassis, see [Virtual Chassis User Guide for EX8200 Switches](#).

## Virtual Chassis Support on EX Series Switches

In an EX Series Virtual Chassis, you can interconnect standalone switches in the following combinations into one logical device, and manage the logical device as a single chassis:

- EX2200 Virtual Chassis, composed of up to four EX2200 switches.
- EX2300 Virtual Chassis, composed of up to four EX2300 switches or up to four EX2300 multigigabit model switches (EX2300-24MP, EX2300-48MP). Starting in Junos OS Release 18.4R1, you can also combine EX2300 multigigabit model switches with other EX2300 switches in the same Virtual Chassis, which operates as a non-mixed Virtual Chassis.  
On EX2300 switches, the Virtual Chassis feature requires license. See [Understanding Software Licenses for EX Series Switches](#).
- EX3300 Virtual Chassis, composed of up to ten EX3300 switches.
- EX3400 Virtual Chassis, composed of up to ten EX3400 switches.
- EX4200 Virtual Chassis, composed of up to ten EX4200 switches.
- EX4300 Virtual Chassis, composed of up to ten EX4300 switches, including multigigabit models (EX4300-48MP). An EX4300 Virtual Chassis operates as a non-mixed Virtual Chassis if it is composed of only EX4300 multigigabit model switches, or composed of any combination of any other EX4300 switches excluding the multigigabit models. An EX4300 Virtual Chassis operates as a

mixed EX4300 Virtual Chassis if it is composed of EX4300 multigigabit model (EX4300-48MP) switches mixed with any other EX4300 model switches.

- EX4400 Virtual Chassis, composed of up to ten EX4400 switches (any EX4400 models, including EX4400 multigigabit models)
- EX4500 Virtual Chassis, composed of up to ten EX4500 switches
- EX4550 Virtual Chassis, composed of up to ten EX4550 switches
- EX4600 Virtual Chassis, composed of up to ten EX4600 switches.
- EX4650 Virtual Chassis, composed of up to two or up to four EX4650-48Y switches.  
Starting in Junos OS Release 19.3R1, an EX4650-48Y Virtual Chassis can have up to two member switches.  
Starting in Junos OS Release 20.1R1, an EX4650-48Y Virtual Chassis can have up to four member switches.
- Mixed EX4200 and EX4500 Virtual Chassis, composed of up to ten total EX4200 and EX4500 switches
- Mixed EX4200 and EX4550 Virtual Chassis, composed of up to ten total EX4200 and EX4550 switches
- Mixed EX4200, EX4500, and EX4550 Virtual Chassis, composed of up to ten total EX4200, EX4500, and EX4550 switches
- Mixed EX4300 and EX4600 Virtual Chassis, composed of up to ten total EX4300 (excluding multigigabit models) and EX4600 switches. The member switches in the primary and backup Routing Engine roles must be EX4600 switches.
- Mixed EX4500 and EX4550 Virtual Chassis, composed of up to ten total EX4500 and EX4550 switches

[Table 1 on page 12](#) and [Table 2 on page 13](#) list the initial Junos OS release for combinations of switches interconnected into an EX Series Virtual Chassis. “N/A” indicates the combination is *not supported*. Switches must be running the same version of Junos OS software to join a Virtual Chassis, although the actual images might be different on different types of switches when mixed hardware models are supported together in a Virtual Chassis.



**Table 1: Minimum Junos OS Release For EX Series Switch Combinations in a Virtual Chassis**

Switch	EX2300 Switch	EX3400 Switch	EX4300 Switch	EX4400 Switch	EX4600 Switch	EX4650
EX2300	15.1X53-D50, 18.1R2 for MP models, 18.4R1 for MP models combined with other models	N/A	N/A	N/A	N/A	N/A
EX3400	N/A	15.1X53-D50	N/A	N/A	N/A	N/A
EX4300	N/A	N/A	13.2X50-D10, 18.2R1 for MP models combined with other models	N/A	13.2X51-D25, excludes EX4300 MP models	N/A
EX4400	N/A	N/A	N/A	21.1R1, 21.2R1 for MP models combined with other models	N/A	N/A
EX4600	N/A	N/A	13.2X51-D25, excludes EX4300 MP models	N/A	13.2X51-D25	N/A
EX4650	N/A	N/A	N/A	N/A	N/A	19.3R1

**Table 2: Minimum Junos OS Release for Legacy EX Series Switch Combinations in a Virtual Chassis**

Switch	EX2200 Switch	EX3300 Switch	EX4200 Switch	EX4500 Switch	EX4550 Switch
EX2200	12.2R1	N/A	N/A	N/A	N/A
EX3300	N/A	11.3R1	N/A	N/A	N/A
EX4200	N/A	N/A	9.0R1	11.1R1	12.2R1
EX4500	N/A	N/A	11.1R1	11.1R1	12.2R1
EX4550	N/A	N/A	12.2R1	12.2R1	12.2R1

## Basic Configuration of EX Series Virtual Chassis

Some EX Series switches can only form a Virtual Chassis with the same type of switches, while some can connect with other types of switches into a mixed-mode Virtual Chassis. See ["Understanding Mixed EX Series and QFX Series Virtual Chassis" on page 39](#) for details on the different types of switches that can be mixed in a Virtual Chassis.

You set up an EX Series Virtual Chassis by configuring Virtual Chassis ports (VCPs) on the member switches, and interconnecting the switches using the VCPs. VCPs are responsible for passing all data and control traffic between member switches in the Virtual Chassis. EX Series switches have one or more of the following VCP options:

- Network or uplink ports that you can configure into VCPs.
- Default-configured VCPs, which are configured as VCPs in the default factory configuration, but you can alternatively convert them into network or uplink ports if desired and reconvert them back into VCPs if needed.
- Dedicated VCPs, which you can only use as VCPs.

See *Virtual Chassis Port Options* for details on which ports on different EX Series switches can be VCPs.

You can increase the VCP bandwidth between any two member switches by connecting multiple VCP links between the switches. When multiple VCP links interconnect the same two member switches, the links automatically form a Link Aggregation Group (LAG) bundle if they have the same speeds. For example, if you have four 40-Gbps links configured as VCPs between two member switches, the four links form a LAG with 160 Gbps of bandwidth. Similarly, connecting two 10-Gbps links configured as

VCPs between two member switches creates a LAG with 2 member links at 20 Gbps total bandwidth. However, 10-Gbps and 40-Gbps links configured as VCPs between two Virtual Chassis member switches cannot be member links of the same VCP LAG.

Within a single wiring closet, you can add a new member switch to a Virtual Chassis by cabling the member switch into the Virtual Chassis using supported VCP links. You can also easily expand a Virtual Chassis configuration beyond a single wiring closet or over a longer distance by connecting member switches together using uplink ports that are supported as VCPs.

You can set up a Virtual Chassis using either a nonprovisioned or a preprovisioned configuration. If you want to deterministically control the role and member ID assigned to each member switch, use a preprovisioned configuration. Virtual Chassis member switches can have one of three roles: primary Routing Engine, backup Routing Engine, or linecard role. In some combinations of switches in a Virtual Chassis, we recommend or require that you configure certain switches into the Routing Engine roles. See *Understanding Virtual Chassis Components* for more information about Virtual Chassis roles.

You can simplify adding switches to a preprovisioned configuration by using the automatic VCP conversion feature (see *Automatic Virtual Chassis Port (VCP) Conversion*), which automatically converts uplink ports into VCPs on the member switches on both sides of the new VCP links as you cable them. This Virtual Chassis expansion method is also called *autoprovisioning*.

## EX2200 Switches in a Virtual Chassis

You can connect up to four EX2200 and EX2200-C switches into an EX2200 Virtual Chassis. EX2200 switches can't be mixed in a Virtual Chassis with any other type of switches.

You can configure and use any EX2200 1-Gigabit Ethernet optical interfaces as VCPs. All RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T RJ-45 transceivers, can also be configured into VCPs. You can configure up to eight 1-Gbps interfaces configured as VCPs into a single Link Aggregation Group (LAG) bundle, so in an EX2200 Virtual Chassis, you can have a VCP LAG with a bandwidth of up to 8 Gigabit Ethernet.

EX2200 Virtual Chassis do not support other high availability features such as Graceful Routing Engine switchover (GRES), Nonstop bridging (NSB), Nonstop active routing (NSR), fast failover, and Nonstop software upgrade (NSSU).

You configure, monitor, and maintain an EX2200 Virtual Chassis is configured and managed similarly to other EX Series Virtual Chassis. See the following for details on configuring a Virtual Chassis with EX2200 switches:

- ["Configuring an EX2200 Virtual Chassis \(CLI Procedure\)" on page 235](#)

## EX2300 Switches in a Virtual Chassis

You can connect up to four EX2300 switches into a Virtual Chassis. You can't combine EX2300 switches into a mixed Virtual Chassis with any other EX Series or QFX Series switches, but you can create a non-mixed Virtual Chassis with different models of EX2300 switches as follows:

- Any combination of up to four EX2300 and EX2300-C switches.
- Any combination of up to four EX2300 multigigabit model (EX2300-24MP and EX2300-48MP) switches.
- (Starting in Junos OS Release 18.4R1) Any combination of up to four EX2300 multigigabit model switches and other EX2300 or EX2300-C model switches. You do not need to set mixed mode.

**NOTE:** In Junos OS releases prior to 18.4R1, you cannot mix EX2300 multigigabit model switches with other EX2300 or EX2300-C model switches in an EX2300 Virtual Chassis.

EX2300 switches do not have default or dedicated VCP ports, but you can configure the 10-Gigabit Ethernet uplink ports as VCPs, and use those to interconnect the switches into a Virtual Chassis.

You connect EX2300 switches or EX2300 multigigabit switches into an EX2300 Virtual Chassis by configuring uplink ports as VCPs and using SFP+ transceivers. The uplink ports on EX2300 switches also support SFP transceivers, but you can't use SFP transceivers on uplink ports to form an EX2300 Virtual Chassis.

In all EX2300 Virtual Chassis, you can configure any EX2300 switch model into any member switch role (primary Routing Engine, backup Routing Engine, or linecard).

On EX2300 switches, the Virtual Chassis feature requires license. See [Understanding Software Licenses for EX Series Switches](#).

You configure, monitor, and maintain an EX2300 Virtual Chassis in a similar way as other EX Series and QFX Series Virtual Chassis. See the following for details on configuring or changing the members in an EX2300 Virtual Chassis:

- *Configuring an EX2300, EX3400, or EX4300 Virtual Chassis*
- *Adding a New Switch to an Existing EX2300, EX3400, or EX4300 Virtual Chassis*
- *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## EX3300 Switches in a Virtual Chassis

You can interconnect up to ten EX3300 switches into a Virtual Chassis. EX3300 switches can't be mixed in a Virtual Chassis with any other type of switches.

Uplink ports 2 and 3 on EX3300 switches are configured as VCPs by default. You can change this default configuration or configure any of the other uplink ports as VCPs.

You configure, monitor, and maintain an EX3300 Virtual Chassis in a similar way as other EX Series Virtual Chassis. See the following for details on configuring or expanding a Virtual Chassis with EX3300 switches:

- ["Configuring an EX3300 Virtual Chassis \(CLI Procedure\)" on page 240](#)
- ["Adding a New Switch to an Existing EX3300 Virtual Chassis \(CLI Procedure\)" on page 261](#)
- *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## EX3400 Switches in a Virtual Chassis

You can connect up to ten of any models of EX3400 switches into a Virtual Chassis. EX3400 switches can't be in a Virtual Chassis with any other EX Series or QFX Series switches.

By default, the QFSP+ uplink ports on EX3400 switches are configured as VCPs, or you can configure any SFP+ uplink module ports on EX3400 switches as VCPs. You can't use uplink ports with SFP transceivers as VCPs to connect EX3400 switches into a Virtual Chassis.

You configure, monitor, and maintain an EX3400 Virtual Chassis in a similar way as other EX Series and QFX Series Virtual Chassis. See the following for details on configuring or changing the members in an EX3400 Virtual Chassis:

- *Configuring an EX2300, EX3400, or EX4300 Virtual Chassis*
- *Adding a New Switch to an Existing EX2300, EX3400, or EX4300 Virtual Chassis*
- *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

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

You can interconnect up to ten EX4200, EX4500, and EX4550 switches into a Virtual Chassis. You must set the Virtual Chassis to mixed mode if it is composed of EX4200 switches with EX4500 or EX4550 switches, but a Virtual Chassis with only one type of these switches or with only EX4500 and EX4550 switches operates as a non-mixed Virtual Chassis.

EX4200 switches have two built-in dedicated VCPs. EX4500 or EX4550 switches have dedicated VCPs on the Virtual Chassis module, and the EX4500 or EX4550 switch must have the PIC mode set to Virtual Chassis mode to interconnect them into a Virtual Chassis. You can also configure any SFP, SFP+, and XFP uplink ports on all of these switches into VCPs. You can use the dedicated VCPs when the switches are close together, such as in the same wiring closet. Use uplink ports configured as VCPs for switches that located farther away, such as in different wiring closets.

You configure, monitor, and maintain an EX4200, EX4500, or EX4550 Virtual Chassis in a similar way as other EX Series Virtual Chassis. See the following for details on configuring a Virtual Chassis with these switches:

- ["Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)" on page 245](#)
- ["Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)" on page 255](#)

For procedures on adding a new switch to a Virtual Chassis composed of any of these switches in a wiring closet, see:

- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\)](#)
- ["Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\)" on page 269](#)
- ["Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\)" on page 271](#)
- ["Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\)" on page 273](#)

## EX4300 Switches in a Virtual Chassis

You can interconnect up to ten EX4300 switches to form an EX4300 Virtual Chassis. EX4300 switches can form a Virtual Chassis composed entirely of EX4300 switches (not multigigabit models) *or* entirely of EX4300 multigigabit model (EX4300-48MP) switches as a non-mixed Virtual Chassis.

You can also connect EX4300 multigigabit model switches and other EX4300 model switches together into an EX4300 Virtual Chassis by configuring all of the member switches into mixed mode. In this case, you must also configure the non-multigigabit EX4300 member switches with a special option (ieee-clause-82) when setting mixed mode. The member switches in the Routing Engine role must be multigigabit model switches, and you must configure the non-multigigabit EX4300 switches into linecard role. Multigigabit model EX4300 switches can't be in a mixed Virtual Chassis with any other types of switches.

**NOTE:** If you remove an EX4300 member switch from a mixed EX4300 Virtual Chassis with multigigabit model members, you must disable ieee-clause-82 port mode on the switch if you want to reconfigure it as a standalone switch or use it in any other type of mixed Virtual Chassis or any non-mixed Virtual Chassis. Otherwise, the VCPs will not connect with other members in the new Virtual Chassis.

You can have any EX4300 switches except multigigabit models in the linecard role in a mixed Virtual Chassis with EX4600 switches or QFX5100 switches. For more details on EX4300 member switches in a mixed Virtual Chassis, see *Understanding Mixed EX Series and QFX Series Virtual Chassis*.

**NOTE:** EX4300 switches (excluding multigigabit models) can also be included as leaf nodes in a Virtual Chassis Fabric (VCF). For information on EX4300 switches in a VCF, see *Virtual Chassis Fabric Overview*.

On EX4300 switches excluding the multigigabit models, all 40-Gigabit Ethernet QSFP+ optical ports are configured as VCPs by default, and you can also configure any 10-Gigabit Ethernet uplink module ports into VCPs as needed. As a result, an EX4300 Virtual Chassis can have either 40-Gbps or 10-Gbps VCP links, or a combination of both. You can easily add new switches to a Virtual Chassis whether the switch is installed in the same building or at a different site because the ports available to be used as VCPs are long-distance optical ports.

On EX4300 multigigabit models, the four 40-Gigabit Ethernet QSFP+ ports on the rear panel are dedicated VCPs. These are the only ports on EX4300 multigigabit model switches that you can use as VCPs, so any EX4300 Virtual Chassis that contains EX4300 multigigabit model switches can have only have VCP links that are 40-Gbps.

You configure, monitor, and maintain an EX4300 Virtual Chassis in a similar way as other EX Series and QFX Series Virtual Chassis. See the following for details on configuring and changing the members in an EX4300 Virtual Chassis:

- *Configuring an EX2300, EX3400, or EX4300 Virtual Chassis*
- *Adding a New Switch to an Existing EX2300, EX3400, or EX4300 Virtual Chassis*
- *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## EX4400 Switches in a Virtual Chassis

You can interconnect up to ten EX4400 switches to form an EX4400 Virtual Chassis. All the member switches must be EX4400 switches (including EX4400 multigigabit models or any other EX4400 models); you can't mix EX4400 switches in a Virtual Chassis with any other type of switches.

You can use any of the EX4400 switch models in either the Routing Engine or linecard role in an EX4400 Virtual Chassis.

On EX4400 switches, by default, each of the two 100-Gigabit Ethernet QSFP28 ports on the rear panel are set as two logical 50-Gbps VCPs, forming four logical 50-Gbps VCP interfaces. You can't use any other ports on EX4400 switches as VCPs. The default VCPs are in PIC slot 1 on EX4400 switches, so the logical VCP interfaces are named vcp-255/1/0 through vcp-255/1/3.

If you don't need to use these ports as VCPs, you can convert them into network ports using the `request virtual-chassis mode network-port` command. In network port mode, you can use these ports as 100 Gigabit Ethernet network ports, or channelize them into four 25-Gigabit Ethernet or four 10-Gigabit Ethernet network ports. To disable network port mode and return these ports to their default settings as VCPs, use the `request virtual-chassis mode network-port disable <reboot>` command. You must reboot the switch for a network port mode change to take effect.

If you disable one of these ports as a VCP using the `request virtual-chassis vc-port delete` command, that action disables the port as a VCP but doesn't change the port to network port mode. You must use the `request virtual-chassis mode network-port` command to use the port as a network port. Also, specifying to disable `vcp-255/1/0` disables both logical ports 0 and 1 (`vcp-255/1/0` and `vcp-255/1/1`), and specifying to disable `vcp-255/1/2` disables both logical ports 2 and 3 (`vcp-255/1/2` and `vcp-255/1/3`).

**NOTE:** You can't use one 100-Gigabit Ethernet port as a VCP and the other as a network port. The two ports must both simultaneously be used in VCP mode or network port mode. Use the `show virtual-chassis mode` command to see the current port mode and what the port mode will be the next time the switch is rebooted. The two values might be different if you changed the mode but have not rebooted the switch yet.

Aside from the VCP differences from other switches, you configure, monitor, and maintain an EX4400 Virtual Chassis in a similar way as other EX Series and QFX Series Virtual Chassis. See the following for details on configuring and changing the members in an EX4400 Virtual Chassis:

- *Configuring an EX2300, EX3400, EX4300, or EX4400 Virtual Chassis*
- *Adding a New Switch to an Existing EX2300, EX3400, EX4300, or EX4400 Virtual Chassis*
- *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## EX4600 Switches in a Virtual Chassis

EX4600 switches can act as member switches in a non-mixed Virtual Chassis—a Virtual Chassis composed entirely of EX4600 switches—and also operate in a mixed Virtual Chassis with EX4300 switches. You can interconnect EX4300 switches (except multigigabit models, EX4300-48MP) with EX4600 switches as a mixed EX4600 Virtual Chassis.

You can interconnect up to ten EX4600 switches or a combination of up to ten EX4600 and EX4300 switches into an EX4600 Virtual Chassis. However, in a mixed Virtual Chassis with EX4600 and EX4300 switches, EX4600 switches must be in the primary and backup Routing Engine roles, and EX4300 member switches must be in the linecard role. As a result, at least two of the member switches in a mixed EX4600 Virtual Chassis must be EX4600 switches in the primary and backup Routing Engine roles. See *Understanding Virtual Chassis Components* for more information about Virtual Chassis member roles.



EX4600 switches do not have any ports that are configured into VCPs by default, but you can configure any 40-Gigabit Ethernet QSFP+ and 10-Gigabit Ethernet SFP+ optical ports on an EX4600 switch into VCPs. Because EX4300 switches support similar VCP options, both non-mixed and mixed EX4600 Virtual Chassis can have 40-Gbps VCP links, 10-Gbps VCP links, or a combination of both.

You configure, monitor, and maintain an EX4600 Virtual Chassis in a similar way as other EX and QFX Series Virtual Chassis. See the following for more details on configuring and changing the members in an EX4600 Virtual Chassis, including a mixed EX4600 Virtual Chassis with EX4300 switches:

- *Configuring EX4600 Switches in a Mixed or Non-Mixed Virtual Chassis*
- *Adding an EX4600 Switch to a Mixed or Non-mixed Virtual Chassis*
- *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

## EX4650 Switches in a Virtual Chassis

Starting in Junos OS Release 19.3R1, you can interconnect up to two EX4650 switches in an EX4650 Virtual Chassis. The two member switches must be in the primary and backup Routing Engine roles.

Starting in Junos OS Release 20.1R1, you can interconnect up to four EX4650 switches in an EX4650 Virtual Chassis. You should configure two member switches into the primary and backup Routing Engine roles, and the remaining member switches into the linecard role.

See *Understanding Virtual Chassis Components* for more information about Virtual Chassis member roles.

EX4650 switches can't be combined with any other type of switches in a Virtual Chassis.

EX4650 switches do not have dedicated or default-configured VCPs, but you can set any of the 40-Gigabit Ethernet QSFP+ or 100-Gigabit Ethernet QSFP28 uplink ports on the front panel (non-channelized ports 48 through 55) as VCPs. You can't use any of the other ports (network ports 0 through 47) as VCPs. Running the `request virtual-chassis vc-port set` command on the network ports doesn't fail, but they will not function properly as VCPs.

An EX4650 Virtual Chassis operates the same as a QFX5120 Virtual Chassis, and you configure, monitor, and maintain it the same way as a QFX Series Virtual Chassis. See the following for more details on configuring and changing the members in an EX4650 Virtual Chassis:

- *Configuring an EX4650 or a QFX Series Virtual Chassis*
- *Adding a New Switch to an Existing EX4650 or QFX Series Virtual Chassis*
- *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*

Release History Table

Release	Description
20.1R1	Starting in Junos OS Release 20.1R1, an EX4650-48Y Virtual Chassis can have up to four member switches.
19.3R1	Starting in Junos OS Release 19.3R1, an EX4650-48Y Virtual Chassis can have up to two member switches.
18.4R1	Starting in Junos OS Release 18.4R1, you can also combine EX2300 multigigabit model switches with other EX2300 switches in the same Virtual Chassis, which operates as a non-mixed Virtual Chassis.

RELATED DOCUMENTATION

<i>Virtual Chassis Overview for Switches</i>
<i>Understanding Virtual Chassis Components</i>
<i>Understanding Virtual Chassis Port Link Aggregation</i>

Understanding Virtual Chassis Components

IN THIS SECTION

- [Maximum Switch Support | 22](#)
- [Virtual Chassis Ports \(VCPs\) | 28](#)
- [Primary Routing Engine Role | 33](#)
- [Backup Routing Engine Role | 34](#)
- [Linecard Role | 35](#)
- [Member Switch and Member ID | 36](#)
- [Primary-role Priority | 37](#)
- [Virtual Chassis Identifier \(VCID\) | 37](#)
- [Nonvolatile Storage in a Virtual Chassis | 38](#)

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

- An EX Series Virtual Chassis is a supported combination of standalone EX Series switches interconnected and managed as a single chassis. This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis.

See *Understanding EX8200 Virtual Chassis Components* for information about EX8200 Virtual Chassis.

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

- A QFX Series Virtual Chassis is a supported combination of standalone QFX3500, QFX3600, QFX5100, QFX5110, QFX5120, or QFX5200 switches interconnected and managed as a single chassis. EX4650 Virtual Chassis operate the same as QFX5120 Virtual Chassis, so most of the information in this topic about QFX Series Virtual Chassis in general also applies to an EX4650 Virtual Chassis, with a few platform-specific support differences.

**NOTE:** EX4300 switches (excluding multigigabit models [EX4300-48MP]) can also be interconnected into a mixed Virtual Chassis with QFX3500, QFX3600, and QFX5100 switches.

This topic does not describe Virtual Chassis Fabric components. Instead, see *Understanding Virtual Chassis Fabric Components*.

## Maximum Switch Support

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

### Maximum Number of Switches in an EX Series Virtual Chassis

[Table 3 on page 23](#) lists the maximum number of member switches supported in an EX Series Virtual Chassis by Junos OS release.

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

Type of EX Series Virtual Chassis	Maximum Member Switches by Junos OS Release
EX2200 Virtual Chassis	12.2R1—Initial release. Up to 4 EX2200 member switches.
EX2300 Virtual Chassis	<p>15.1X53-D50—Initial release. Up to 4 EX2300 member switches.</p> <p>18.1R2—Up to 4 multigigabit EX2300 (EX2300-24MP and EX2300-48MP) member switches.</p> <p>18.4R1—Starting in Junos OS Release 18.4R1, up to 4 of any model EX2300 member switches (including multigigabit models and any other EX2300 switches) can be combined in the same Virtual Chassis.</p>
EX3300 Virtual Chassis	<p>11.3R1—Initial release. Up to 6 EX3300 member switches.</p> <p>12.2R1—Starting in Junos OS Release 12.2R1, an EX3300 Virtual Chassis can support up to 10 EX3300 member switches.</p>
EX3400 Virtual Chassis	15.1X53-D50—Initial release. Up to 10 EX3400 member switches.
EX4200 Virtual Chassis	9.0R1—Initial release. Up to 10 EX4200 member switches.

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

Type of EX Series Virtual Chassis	Maximum Member Switches by Junos OS Release
EX4300 Virtual Chassis	<p>13.2X50-D10—Initial release. Up to 10 EX4300 member switches.</p> <p>13.2X50-D20—Starting in Junos OS Release 13.2X50-D20, a mixed QFX Series Virtual Chassis or VCF can also contain EX4300 switches.</p> <p>18.2R1—Starting in Junos OS Release 18.2R1 with the introduction of EX4300 multigigabit model switches (EX4300-48MP), an EX4300 Virtual Chassis can contain up to 10 EX4300 multigigabit model switches as a non-mixed Virtual Chassis or a combination of EX4300 multigigabit model switches with other EX4300 switches as a mixed EX4300 Virtual Chassis.</p>
EX4400 Virtual Chassis	<p>21.1R1—Initial release. Up to 10 EX4400 member switches.</p> <p>21.2R1—Starting in Junos OS Release 21.2R1, an EX4400 Virtual Chassis can also include EX4400 multigigabit model switches (EX4400-24MP and EX4400-48MP).</p>
EX4500 Virtual Chassis	<p>11.1R1—Initial release. Support for up to 2 EX4500 member switches.</p> <p>11.4R1—Support for up to 10 EX4500 member switches.</p>
EX4550 Virtual Chassis	<p>12.2R1—Initial release. Support for up to 10 EX4550 member switches.</p>
EX4600 Virtual Chassis	<p>13.2X51-D25—Initial release. Support for up to 10 EX4600 member switches.</p>

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

Type of EX Series Virtual Chassis	Maximum Member Switches by Junos OS Release
EX4650 Virtual Chassis	<p>19.3R1—Initial release. Up to 2 EX4650 switches in Routing Engine roles only.</p> <p>20.1R1—Starting in Junos OS Release 20.1R1, an EX4650 Virtual Chassis can have up to 4 members.</p>
Mixed EX4200 and EX4500 Virtual Chassis	<p>11.1R1—Initial release. Up to 2 EX4500 member switches and up to 8 EX4200 member switches.</p> <p>11.2R1—Up to 9 EX4200 member switches.</p> <p>11.4R1—Up to 9 EX4500 member switches.</p>
Mixed EX4200 and EX4550 Virtual Chassis	<p>12.2R1—Initial release. Up to 10 total EX4200 and EX4550 member switches.</p>
Mixed EX4200, EX4500, and EX4550 Virtual Chassis	<p>12.2R1—Initial release. Up to 10 total EX4200, EX4500, and EX4550 member switches.</p>
Mixed EX4300 and EX4600 Virtual Chassis	<p>13.2X51-D25—Initial release. Up to 10 total EX4300 and EX4600 member switches. EX4600 member switches must assume Routing Engine role.</p> <p><b>NOTE:</b> EX4300 multigigabit model (EX4300-48MP) switches are not supported in a mixed Virtual Chassis with EX4600 switches.</p>
Mixed EX4500 and EX4550 Virtual Chassis	<p>12.2R1—Initial release. Up to 10 total EX4500 and EX4550 switches.</p>

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

Type of EX Series Virtual Chassis	Maximum Member Switches by Junos OS Release
EX9200 Virtual Chassis	<p>13.2R2—Initial release. Up to 2 EX9200 switches.</p> <p><b>NOTE:</b> We have phased out support for EX9200 switches in a Virtual Chassis as of Junos OS Release 17.1R1. For deployments with EX9200 switches, we recommend planning or moving to MC-LAG or Junos Fusion Enterprise architectures instead of using a Virtual Chassis configuration.</p>

### Maximum Number of Switches in a QFX Series Virtual Chassis (Including Mixed Virtual Chassis with EX Series Switches)

Table 4 on page 26 lists the maximum number of member switches supported in a QFX Series Virtual Chassis by Junos OS release, including mixed QFX Series Virtual Chassis with EX Series switch members.

**Table 4: Maximum Member Switch Support for QFX Series Virtual Chassis by Junos OS Release**

Type of QFX Series Virtual Chassis	Maximum Member Switches by Junos OS Release
<p>QFX3500 or QFX3600 Virtual Chassis:</p> <ul style="list-style-type: none"> <li>Only QFX3500 and QFX3600 switches in any combination.</li> </ul>	13.2X50-D15—Initial release. Up to 10 total member switches.
<p>QFX3500 or QFX3600 mixed-mode Virtual Chassis:</p> <ul style="list-style-type: none"> <li>QFX3500 and QFX3600 switches in Routing Engine role with any combination of QFX3500, QFX3600, and EX4300 switches (excluding EX4300 multigigabit models) in linecard role.</li> </ul>	13.2X51-D20—Up to 10 total member switches.

**Table 4: Maximum Member Switch Support for QFX Series Virtual Chassis by Junos OS Release**  
*(Continued)*

Type of QFX Series Virtual Chassis	Maximum Member Switches by Junos OS Release
<p>QFX5100 Virtual Chassis:</p> <ul style="list-style-type: none"> <li>Only QFX5100 switches.</li> </ul>	<p>13.2X51-D20—Initial release. Up to 10 member switches (except QFX5100-96S).</p> <p>13.2X51-D20—Initial release for QFX5100-96S. Up to 4 member switches.</p> <p>13.2X53-D25—Prior to Junos OS Release 13.2X51-D25, only up to 4 member switches can be in a QFX5100 Virtual Chassis. Starting in Junos OS Release 13.2X51-D25, you can include up to 10 QFX5100-96S switches in a mixed or non-mixed QFX5100 Virtual Chassis.</p>
<p>QFX5100 mixed-mode Virtual Chassis:</p> <ul style="list-style-type: none"> <li>QFX5100 switches in Routing Engine role with any combination of QFX5100, QFX3500, QFX3600, and EX4300 switches (excluding EX4300 multigigabit models) in linecard role.</li> </ul>	<p>13.2X51-D20—Initial release. Up to 10 total member switches (except QFX5100-96S).</p> <p>13.2X53-D25—Up to 10 total member switches (including QFX5100-96S).</p>
<p>QFX5110 Virtual Chassis:</p> <ul style="list-style-type: none"> <li>QFX5110 switches in Routing Engine role with any combination of supported QFX5110 and QFX5100 switches in linecard role.</li> </ul>	<p>17.3R1—Initial release. Up to 10 member switches.</p>
<p>QFX5120 Virtual Chassis:</p>	<p>19.3R1—Initial release on QFX5120-48Y switches. Up to 2 member switches, both in Routing Engine role.</p> <p>20.2R1—Initial release on QFX5120-48T switches. Up to 2 member switches, both in Routing Engine role.</p> <p>20.3R1—Initial release on QFX5120-32C switches. Up to 2 member switches, both in Routing Engine role.</p>
<p>QFX5200 Virtual Chassis—</p> <ul style="list-style-type: none"> <li>Only QFX5200 switches.</li> </ul>	<p>17.3R2 and 17.4R1—Initial release. Up to 3 member switches.</p>



## Virtual Chassis Ports (VCPs)

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

### Virtual Chassis Port Options

Some switches have dedicated VCPs; you can only use these ports as VCPs and you can't reconfigure them as network ports. Dedicated VCPs allow you to interconnect switches into a Virtual Chassis without requiring any additional interface configuration.

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

Most switches have optical or uplink ports that you can also configure as VCPs.

You must configure VCPs to interconnect switches that do not have dedicated or default-configured VCPs or to interconnect switches across greater distances than allowed by a dedicated VCP connection. Otherwise, you can mix any of the supported VCP options among the members of a Virtual Chassis, and we recommend having redundant links between any two members for resiliency or to increase member communication bandwidth. VCPs automatically bundle into a Link Aggregation Group when two or more ports operating at the same speed are configured into VCPs between the same two member switches. See *Understanding Virtual Chassis Port Link Aggregation* for details.

When adding switches to an existing Virtual Chassis or adding new redundant links between existing members, if the automatic VCP conversion feature is enabled, under the right conditions the ports on both sides of the connection will convert into VCPs automatically (see ["Automatic Virtual Chassis Port \(VCP\) Conversion" on page 32](#)).

[Table 5 on page 29](#) summarizes the available VCP options on switches in an EX Series or QFX Series Virtual Chassis. For complete details on where dedicated VCPs, default-configured VCPs, or ports that can be configured as VCPs are located on a switch, and the supported transceivers and cables that you can use for VCP connections on the switch, see the hardware documentation for that type of switch.

Table 5: VCP Options by Switch Type

Switch	Dedicated VCPs	Default VCPs	Ports that can be configured and are supported as VCPs
EX2200	None	None	Any uplink ports  All RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T RJ-45 transceivers
EX2300 (including EX2300 multigigabit models)	None	None	10-Gigabit Ethernet uplink ports with SFP+ transceivers  <b>NOTE:</b> You cannot use ports with SFP transceivers as VCPs on EX2300 switches to form a Virtual Chassis.
EX3300	None	Uplink ports 2 and 3	Any of the 4 uplink ports (ports 0 through 3)
EX3400	None	All QSFP+ uplink ports (PIC slot 1, ports 0 and 1)	Any SFP+ uplink ports  <b>NOTE:</b> You cannot use ports with SFP transceivers as VCPs on EX3400 switches to form a Virtual Chassis.
EX4200	2 ports on the rear panel	None	Any uplink module ports (SFP, SFP+, or XFP) or through an SFP+ port on the EX4200-24F switch  <b>NOTE:</b> You cannot set a 1000BASE-T copper SFP transceiver (EX-SFP-1GE-T) connection as a VCP on EX4200 switches.
EX4300	None	All QSFP+ ports	Any uplink ports installed with SFP+ or QSPF+ transceivers  <b>NOTE:</b> On 32-port EX4300 switches, you can't use the four built-in 10-Gigabit Ethernet SFP+ ports as VCPs.

Table 5: VCP Options by Switch Type *(Continued)*

Switch	Dedicated VCPs	Default VCPs	Ports that can be configured and are supported as VCPs
EX4300 multigigabit models (EX4300-48MP)	4 40-Gbps QSFP+ ports on the rear panel	None	None
EX4400 (Including EX4400 multigigabit models)	None	4 logical 50-Gbps VCP interfaces using the two 100-Gbps QSFP28 ports on the rear panel (PIC slot 1)	None
EX4500 and EX4550	Two ports on the Virtual Chassis module	None	Any SFP, SFP+, or XFP uplink module port  <b>NOTE:</b> You cannot use SFP+ uplink ports installed with 1000BASE-T copper SFP transceivers (EX-SFP-1GE-T) as VCP connections on EX4500 and EX4550 switches.
EX4600	None	None	Any SFP+ and QSFP+ ports
EX4650	None	None	Any of the 40-Gigabit Ethernet or 100-Gigabit QSFP28 ports on the front panel (ports 48 through 55), non-channelized  <b>NOTE:</b> The Junos OS doesn't prevent you from trying to set other ports as VCPs, but they don't operate properly as VCPs.
QFX3500 and QFX3600	None	None	Any non-channelized 40-Gigabit Ethernet QSFP+ interfaces
QFX5100	None	None	Any non-channelized 40-Gigabit Ethernet QSFP+ interfaces

Table 5: VCP Options by Switch Type *(Continued)*

Switch	Dedicated VCPs	Default VCPs	Ports that can be configured and are supported as VCPs
QFX5110	None	None	<p>Any 40-Gigabit Ethernet or 100-Gigabit Ethernet QSFP28 ports</p> <p>Any non-channelized 40-Gigabit Ethernet QSFP+ interfaces</p> <p>Any non-channelized 10-Gigabit Ethernet SFP+ interfaces (on QFX5110 switch models that support these ports)</p>
QFX5120	None	None	<p>(QFX5120-48Y) Any of the eight 40-Gigabit Ethernet or 100-Gigabit Ethernet QSFP+ or QSFP28 ports on the front panel (ports 48 through 55), non-channelized</p> <p>(QFX5120-48T) Any of the six 40-Gigabit Ethernet or 100-Gigabit Ethernet QSFP+ or QSFP28 ports on the front panel (ports 48 through 53), non-channelized</p> <p><b>NOTE:</b> Any ports other than those specified above for QFX5120-48Y and QFX5120-48T switches are not supported as VCPs. The Junos OS CLI doesn't return an error if you try to set other ports as VCPs, but they will not work properly as VCPs.</p> <p>(QFX5120-32C) Any non-channelized network ports (ports 0 through 31) installed with either 40-Gbps QSFP+ or 100-Gbps QSFP28 transceivers</p>
QFX5200	None	None	<p>Any 40-Gigabit Ethernet QSFP+ ports</p> <p>Starting in Junos OS Release 17.3R2-S4, you can also use 100-Gigabit Ethernet QSFP28 ports as VCPs on QFX5200 switches.</p>

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

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

## Automatic Virtual Chassis Port (VCP) Conversion

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

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

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

For ports to be eligible for automatic VCP conversion, you must convert them back into network ports using the `request virtual-chassis vc-port delete` command if they are default-configured VCPs or you previously configured them into VCPs. Switches do not automatically convert VCPs back into network ports when you remove them from a Virtual Chassis and disconnect the links.

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

- Automatic VCP conversion doesn't apply to EX4400 switches in a Virtual Chassis. On these switches, to convert the default VCPs into network ports or convert them from network ports back into VCP ports, you must explicitly set the port mode using the `request virtual-chassis mode network-port` command, and then reboot the switch.
- Starting in Junos OS Releases 15.1R7 and 14.1X53-D47, in EX2200, EX3300, EX4200, EX4500, and EX4550 Virtual Chassis, automatic VCP conversion is disabled by default. If desired, you can enable the feature by configuring the `auto-conversion` statement at the `[edit virtual-chassis]` hierarchy level on the Virtual Chassis.



**CAUTION:** When automatic VCP conversion is enabled in a Virtual Chassis with switches that have dedicated VCPs (EX4200, EX4500, or EX4550 Virtual Chassis), if network or uplink ports are automatically converted into VCPs to create a redundant link with a dedicated VCP connection between the same two Virtual Chassis members, you must reboot the Virtual Chassis to avoid creating a traffic loop within the Virtual

Chassis. (The same issue can happen even if you manually convert the ports into VCPs to create the redundant VCP link with a dedicated VCP link, so you must also reboot the Virtual Chassis in that case to avoid traffic looping.)

- Starting in Junos OS Releases 14.1X53-D47, 17.4R2, 18.1R3, 18.2R2, and 18.3R1 for EX4300, EX4600, QFX Series Virtual Chassis and for any EX4650 and QFX5120 Virtual Chassis (which all have the automatic VCP conversion feature enabled by default), you can choose to disable the feature by configuring `no-auto-conversion` at the `[edit virtual-chassis]` hierarchy level on the Virtual Chassis. To return to the default behavior to re-enable automatic VCP conversion, delete the `no-auto-conversion` statement from the configuration.

## Virtual Chassis Port Link Aggregation Groups

You can increase VCP bandwidth between member switches by configuring multiple links between the same two switches into VCP links. When multiple VCPs interconnect the same two member switches, the links automatically form a Link Aggregation Group (LAG) bundle if the VCP links are the same speed. For example, if you have two 40-Gbps QSFP+ VCP links connected between member switches, the links automatically form a LAG with 80-Gbps total bandwidth. However, 10-Gigabit SFP+ and 40-Gbps QSFP+ VCP links will not become members of the same LAG.

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

## Primary Routing Engine Role

In a Virtual Chassis, each member switch operates in one of two roles, Routing Engine role or linecard role. When in Routing Engine role, a member switch acts as the primary or backup Routing Engine.

The primary Routing Engine member in the Virtual Chassis:

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

In a preprovisioned configuration, the Virtual Chassis primary-role election algorithm determines which member switch in the Routing Engine role acts as the Virtual Chassis primary and which acts as the backup. See *Understanding How the Primary in a Virtual Chassis Is Elected*.

In a configuration that is not preprovisioned, called a *nonprovisioned* configuration, the Virtual Chassis selects the primary and backup using the primary-role priority value and secondary factors in the primary-role election algorithm.

The remaining switches in the Virtual Chassis that are not the primary or backup operate in the linecard role.

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

- In any mixed Virtual Chassis configuration that includes EX4200 switches, EX4500 switches, or EX4550 switches, you can configure any switch in any role in any configuration.
- In a mixed EX4300 Virtual Chassis composed of EX4300 multigigabit model (EX4300-48MP) and other EX4300 model switches, you should always have EX4300 multigigabit model switches in the Routing Engine role.
- In a mixed EX4600 Virtual Chassis with EX4300 switches, EX4600 switches are always in the primary Routing Engine role.
- In a mixed QFX Series Virtual Chassis made up of QFX5100 switches with QFX3500, QFX3600, or EX4300 switches, we recommend configuring QFX5100 switches into the Routing Engine role.
- In a mixed QFX Series Virtual Chassis made up of QFX3500, QFX3600, and EX4300 switches, we recommend configuring QFX3500 or QFX3600 switches into the primary and backup Routing Engine roles.
- In a QFX5110 Virtual Chassis with QFX5110 and QFX5100 switches, we recommend configuring only QFX5110 switches into the Routing Engine role.
- In a two-member EX4650 or QFX5120 Virtual Chassis, configure both member switches into the Routing Engine role as primary and backup member switches only (no linecard role members).

## Backup Routing Engine Role

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

- Maintains a state of readiness to take over the primary Routing Engine role if the primary fails.
- Runs Junos OS for the switches as a backup Routing Engine.
- Synchronizes with the primary in terms of protocol states, forwarding tables, and other information, so that it is prepared to preserve routing information and maintain network connectivity without disruption in case the primary is unavailable.

The Virtual Chassis configuration must have at least two member switches in order to have a backup Routing Engine member.

In a preprovisioned configuration, the Virtual Chassis primary-role election algorithm determines which member switch in the Routing Engine role acts as the Virtual Chassis primary and which acts as the backup. See *Understanding How the Primary in a Virtual Chassis Is Elected*.

In a nonprovisioned configuration, the Virtual Chassis selects the primary and backup member switches using the primary-role priority value and secondary factors in the primary-role election algorithm.

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

- In any mixed Virtual Chassis configuration that includes EX4200 switches, EX4500 switches, or EX4550 switches, you can configure any switch in any role in any configuration.
- In a mixed EX4300 Virtual Chassis composed of EX4300 multigigabit model (EX4300-48MP) and other EX4300 model switches, you should always have EX4300 multigigabit model switches in the primary and backup Routing Engine roles.
- In a mixed EX4600 Virtual Chassis with EX4300 switches, you must use an EX4600 switch in the primary role, and we strongly recommend you also configure an EX4600 switch into the backup role to help the Virtual Chassis remain stable when a routing engine switchover happens.
- In a mixed QFX Series Virtual Chassis made up of QFX5100 switches with QFX3500, QFX3600, or EX4300 switches, we recommend configuring the QFX5100 switches into the Routing Engine role.
- In a mixed QFX Series Virtual Chassis made up of QFX3500, QFX3600, and EX4300 switches, we recommend configuring only QFX3500 or QFX3600 switches into the primary and backup Routing Engine roles.
- In a QFX5110 Virtual Chassis with QFX5110 and QFX5100 switches, we recommend configuring only QFX5110 switches into the Routing Engine role.
- In a two-member EX4650 or QFX5120 Virtual Chassis, configure both member switches into the Routing Engine role as primary and backup member switches only (no linecard role members).

## Linecard Role

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

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

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



In a preprovisioned configuration, you can explicitly configure a member with the linecard role, which means it can't be a primary or backup Routing Engine.

In a nonprovisioned configuration, the members that are not selected as primary or backup operate as linecard members of the Virtual Chassis. The Virtual Chassis selects the primary and backup member switches using the primary-role priority value and secondary factors in the primary-role election algorithm. A switch with a primary-role priority of 0 is always in the linecard role.

In any two-member Virtual Chassis, for high availability you should configure both members into the Routing Engine role, and no members in the linecard role. Otherwise, in a Virtual Chassis with more than two members, any supported switch type can operate in linecard role.

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

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

## Member Switch and Member ID

Each standalone switch that supports Virtual Chassis is a potential member of a Virtual Chassis configuration. When you power on one of those switches, it has a Virtual Chassis member ID that you can see on the front-panel LCD on some switches or in `show virtual-chassis` command output. If the switch is powered on as a standalone switch, its member ID is always 0. When you interconnect the switch into a Virtual Chassis configuration, the primary member switch assigns it a member ID based on various factors such as the order in which the switch was added to the Virtual Chassis or if you defined member IDs based on switch serial numbers in the preprovisioning process.

If the Virtual Chassis configuration previously included a member switch and you physically disconnected or removed that member from the Virtual Chassis configuration, its member ID is not automatically available for assignment as part of the primary's standard sequential member ID assignment. For example, you might have a Virtual Chassis configuration with member 0, member 2, and member 3, because member 1 was removed. When you add another member switch and power it on, the primary assigns ID 4 to it, not ID 1. If you want to reuse a member ID from a member switch that was removed, you can *recycle* the member id (see the `request virtual-chassis recycle` command for details).

The member ID distinguishes the member switches from each other. You use the member ID to:

- assign a primary-role priority value to a member switch.

- configure interfaces for a member switch, similar to specifying a juniper Networks device slot number.
- apply some operational commands to a member switch.
- display status or characteristics of a member switch.

## Primary-role Priority

In a nonprovisioned configuration, you can designate the role (primary or backup Routing Engine role or linecard role) that a member switch assumes by configuring its primary-role priority (a number from 0 through 255). The primary-role priority value is the first consideration in the primary-role election algorithm for selecting the primary of the Virtual Chassis configuration. A switch with a primary-role priority of 0 never assumes the backup or primary Routing Engine role.

When you power on a standalone switch, it has the default primary-role priority value 128. Because it's the only member switch in its own Virtual Chassis configuration, it's also the primary member. When you interconnect a standalone switch to an existing Virtual Chassis configuration (which already has its own primary), we recommend that you explicitly configure the primary-role priority of the members that you want to function as the primary and backup.

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

In a preprovisioned configuration, you can't configure primary-role priority values manually. You assign the role of each member switch, and the Virtual Chassis assigns the primary-role priority automatically based on the assigned role.

## Virtual Chassis Identifier (VCID)

All members of a Virtual Chassis configuration share one Virtual Chassis identifier (VCID). The Virtual Chassis derives this identifier from internal parameters. When you monitor a Virtual Chassis configuration, certain interface views and the `show virtual-chassis` command display the VCID.

## Nonvolatile Storage in a Virtual Chassis

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

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

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

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

**Release History Table**

Release	Description
20.1R1	Starting in Junos OS Release 20.1R1, an EX4650 Virtual Chassis can have up to 4 members.
18.4R1	Starting in Junos OS Release 18.4R1, up to 4 of any model EX2300 member switches (including multigigabit models and any other EX2300 switches) can be combined in the same Virtual Chassis.
18.2R1	Starting in Junos OS Release 18.2R1 with the introduction of EX4300 multigigabit model switches (EX4300-48MP), an EX4300 Virtual Chassis can contain up to 10 EX4300 multigigabit model switches as a non-mixed Virtual Chassis or a combination of EX4300 multigigabit model switches with other EX4300 switches as a mixed EX4300 Virtual Chassis.
17.3R2-S4	Starting in Junos OS Release 17.3R2-S4, you can also use 100-Gigabit Ethernet QSFP28 ports as VCPs on QFX5200 switches.
15.1R7	Starting in Junos OS Releases 15.1R7 and 14.1X53-D47, in EX2200, EX3300, EX4200, EX4500, and EX4550 Virtual Chassis, automatic VCP conversion is disabled by default.
14.1X53-D47	Starting in Junos OS Releases 14.1X53-D47, 17.4R2, 18.1R3, 18.2R2, and 18.3R1 for EX4300, EX4600, QFX Series Virtual Chassis and for any EX4650 and QFX5120 Virtual Chassis (which all have the automatic VCP conversion feature enabled by default), you can choose to disable the feature by configuring no-auto-conversion at the [edit virtual-chassis] hierarchy level on the Virtual Chassis.

13.2X53-D25	Starting in Junos OS Release 13.2X51-D25, you can include up to 10 QFX5100-96S switches in a mixed or non-mixed QFX5100 Virtual Chassis.
13.2X50-D20	Starting in Junos OS Release 13.2X50-D20, a mixed QFX Series Virtual Chassis or VCF can also contain EX4300 switches.
12.2R1	Starting in Junos OS Release 12.2R1, an EX3300 Virtual Chassis can support up to 10 EX3300 member switches.

RELATED DOCUMENTATION

<i>Virtual Chassis Overview for Switches</i>
<i>Understanding EX8200 Virtual Chassis Components</i>
<i>Understanding EX Series Virtual Chassis</i>
<i>Understanding QFX Series Virtual Chassis</i>
<i>Understanding Mixed EX Series and QFX Series Virtual Chassis</i>
<i>Configuring an EX4650 or a QFX Series Virtual Chassis</i>
<i>Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port</i>
<i>Command Forwarding Usage with EX Series and QFX Series Virtual Chassis</i>
<a href="#">Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet   98</a>
<a href="#">Example: Configuring an EX4500 Virtual Chassis with a Primary and Backup in a Single Wiring Closet   105</a>
<a href="#">Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File   168</a>

Understanding Mixed EX Series and QFX Series Virtual Chassis

IN THIS SECTION

- [Mixed and Non-mixed EX Series and QFX Series Virtual Chassis Summary | 41](#)
- [Understanding the Routing Engine Role in a Virtual Chassis With Different Types of Switches | 44](#)
- [Understanding QFX5100 and QFX5110 Switches in a Virtual Chassis | 44](#)
- [Understanding EX4300, QFX3500, QFX3600, and QFX5100 Switches in a Mixed Virtual Chassis | 45](#)

- [Understanding Mixed EX4300 and EX4600 Virtual Chassis | 45](#)
- [Understanding EX4300 Multigigabit and Other EX4300 Model Switches in a Mixed EX4300 Virtual Chassis | 46](#)
- [Understanding EX4200, EX4500, and EX4550 Switches in a Mixed Virtual Chassis | 47](#)

This topic describes the requirements for a mixed Virtual Chassis.

A *mixed Virtual Chassis* includes two or more types of EX Series switches, two or more types of QFX Series switches, or a supported combination of EX and QFX Series switches, where architectural differences require the Virtual Chassis to be configured into mixed mode for the switches to interoperate.

A Virtual Chassis composed of all the same type of switch can usually operate as a *non-mixed Virtual Chassis*, which does not require you to set mixed mode. However, the following Virtual Chassis that include different models of the same product must operate in mixed mode due to architecture differences between the different models:

- An EX4300 Virtual Chassis composed of EX4300 multigigabit model (EX4300-48MP) switches mixed with any other EX4300 model switches

**NOTE:** An EX4300 Virtual Chassis operates as a non-mixed Virtual Chassis if it is composed of only EX4300 multigigabit model switches, or composed of any combination of any other EX4300 switches (excluding the multigigabit models).

The following combinations of different switch types can be interconnected into a Virtual Chassis that does *not* require you to set mixed mode because the switches can run the same software image when in a Virtual Chassis:

- An EX Series Virtual Chassis composed of only EX4500 and EX4550 switches
- An EX2300 Virtual Chassis composed of any models of EX2300 and EX2300 multigigabit switches

**NOTE:** Junos OS releases prior to 18.4R1 support forming an EX2300 Virtual Chassis using only EX2300 multigigabit switches or only EX2300 switches that are not multigigabit model switches. Starting in Junos OS Release 18.4R1, EX2300, EX2300-C, and EX2300 multigigabit switches can all be combined in the same non-mixed Virtual Chassis.

- An EX4400 Virtual Chassis composed of any models of EX4400 and EX4400 multigigabit switches

- A QFX Series Virtual Chassis composed of only QFX3500 and QFX3600 switches
- A QFX5110 Virtual Chassis composed of QFX5110 and supported QFX5100 switches

QFX5200, QFX5120, EX4650, EX4400, EX3400, EX3300, EX2300, and EX2200 switches cannot be mixed with any other models of switches in a Virtual Chassis.

## Mixed and Non-mixed EX Series and QFX Series Virtual Chassis Summary

Table 6 on page 41 provides a high-level overview of the EX Series and QFX Series switches allowed in the Routing Engine and line-card roles of supported mixed and non-mixed Virtual Chassis configurations. Any supported mixed or non-mixed combination of switches can be configured as a nonprovisioned or preprovisioned Virtual Chassis.

**Table 6: Virtual Chassis Summary**

Category	Allowed Routing Engine Members	Allowed Line Card Members
Non-mixed	QFX5200	QFX5200
	QFX5120-48Y or QFX5120-48T or QFX5120-32C (2 switches of the same model only)	None, only 2 members of the same model switch are supported in any QFX5120 Virtual Chassis, and both must be in Routing Engine role
	QFX5110	QFX5110 QFX5100 (with a "-qfx-5e-" Junos OS image)
	QFX5100	QFX5100
	QFX3600 QFX3500	QFX3600 QFX3500
	EX4650	None if only 2 members in the Virtual Chassis (both must be in Routing Engine role), or only other EX4650 switches otherwise

Table 6: Virtual Chassis Summary (*Continued*)

Category	Allowed Routing Engine Members	Allowed Line Card Members
	EX4600	EX4600
	EX4400 (including multigigabit and any other models)	EX4400 (including multigigabit and any other models)
	EX4300 multigigabit models (EX4300-48MP) only	EX4300 multigigabit models (EX4300-48MP) only
	EX4300 (any models except multigigabit models)	EX4300 (any models except multigigabit models)
	EX4200	EX4200
	EX4500	EX4500
	EX4550	EX4550
	EX3400	EX3400
	EX3300	EX3300
	(Prior to Junos OS Release 18.4R1) EX2300 multigigabit models (EX2300-24MP and EX2300-48MP) only, or EX2300 and EX2300-C (excluding multigigabit models) only	(Prior to Junos OS Release 18.4R1) EX2300 multigigabit models (EX2300-24MP and EX2300-48MP) only, or EX2300 and EX2300-C (excluding multigigabit models) only

Table 6: Virtual Chassis Summary *(Continued)*

Category	Allowed Routing Engine Members	Allowed Line Card Members
	(Starting with Junos OS Release 18.4R1) EX2300, EX2300-C, and EX2300 multigigabit models (EX2300-24MP and EX2300-48MP) (any models in any combination)	(Starting with Junos OS Release 18.4R1) EX2300, EX2300-C, and EX2300 multigigabit models (EX2300-24MP and EX2300-48MP) (any models in any combination))
	EX2200	EX2200
Mixed	QFX5100	QFX5100 QFX3600 QFX3500 EX4300 (any models except multigigabit models)
	QFX3600 QFX3500	QFX3600 QFX3500 EX4300 (any models except multigigabit models)
	EX4600	EX4600 EX4300 (any models except multigigabit models)
	EX4300 multigigabit models (EX4300-48MP)	EX4300 (any models including multigigabit models)
	EX4200, EX4500, EX4550 (any of these switches in any mixed combination)	EX4200, EX4500, EX4550 (any of these switches in any mixed combination)



## Understanding the Routing Engine Role in a Virtual Chassis With Different Types of Switches

When you have different types of switches in a Virtual Chassis, the combination of switches you are interconnecting determines which switches should be in the primary Routing Engine role. We also recommend always configuring the same type of switch into the primary and backup Routing Engine roles, to ensure that the switch operating as the primary remains the same type of switch in the event of a switchover.

- In a Virtual Chassis with QFX5110 and QFX5100 switches, which is considered to be a non-mixed QFX5110 Virtual Chassis, we recommend using QFX5110 switches in the primary or backup Routing Engine roles, and you can use QFX5110 or supported QFX5100 switches for the remaining members in line-card role.
- In a mixed Virtual Chassis with QFX5100 switches combined with QFX3600, QFX3500, or EX4300 switches, you should use QFX5100 switches in the primary Routing Engine role, and you can use QFX5100, QFX3600, QFX3500, or EX4300 switches in the line-card role.
- In a mixed Virtual Chassis with QFX3600 or QFX3500 switches with EX4300 switches, you should use QFX3500 or QFX3600 switches in the Routing Engine role, and you can use QFX3600, QFX3500, or EX4300 switches in the line-card role.
- In a mixed EX4300 and EX4600 Virtual Chassis, EX4600 switches automatically assume the primary and backup Routing Engine roles, and you can use EX4300 or EX4600 switches in the line-card role. (EX4600 switches can be in a mixed Virtual Chassis with any EX4300 model switches except multigigabit models.)
- In a mixed EX4300 Virtual Chassis with EX4300 multigigabit model (EX4300-48MP) switches and any other models of EX4300 switches, you must use EX4300 multigigabit models in the Routing Engine role, and you can use any EX4300 switches in the line-card role.
- In a mixed Virtual Chassis that includes any combination of EX4200 switches, EX4500 switches, and EX4550 switches, any switch can be configured in any role.

In most mixed Virtual Chassis, you must configure your Virtual Chassis to ensure a switch that supports operating as a primary Routing Engine assumes the primary Routing Engine role. Without user configuration, any switch might assume the primary or backup Routing Engine role, with the exception of EX4300 switches in an EX4600 or QFX Series Virtual Chassis, which can never assume the Routing Engine role.

## Understanding QFX5100 and QFX5110 Switches in a Virtual Chassis

Up to ten QFX5100 and QFX5110 switches can be interconnected using Virtual Chassis ports (VCPs) to form a *QFX5110 Virtual Chassis*, which is considered to be a non-mixed Virtual Chassis because both types of switches can run the same software image, and you do not need to configure mixed mode. A

QFX5110 Virtual Chassis can contain QFX5110 and supported QFX5100 switches in any combination, but we recommend that only QFX5110 switches be in the primary and backup Routing Engine roles, and QFX5100 switches only be configured into line-card role.

**NOTE:** A QFX5110 Virtual Chassis with QFX5100 switches can only be set up using QFX5110 and QFX5100 switches running the same Junos OS image that includes “-qfx-5e-” in the Junos OS software package filename downloaded from the Software Center. QFX5100 switches running a Junos OS image filename that includes “-qfx-5-” must first be upgraded to the “-qfx-5e-” image to join a QFX5110 Virtual Chassis. (See *Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric*.)

QFX5100 switches can also be interconnected into a mixed Virtual Chassis with QFX3500, QFX3600, and EX4300 switches. See the next section for more information on a mixed QFX5100 Virtual Chassis.

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

A combination of up to ten EX4300 (excluding multigigabit models), QFX3500, QFX3600, and QFX5100 switches can be interconnected using VCPs to form a mixed QFX Series Virtual Chassis.

In a mixed QFX Series Virtual Chassis that includes QFX5100 switches, only QFX5100 switches should be configured into the Routing Engine role, and the remaining members can be any combination of EX4300, QFX3500, QFX3600, or QFX5100 switches configured into line-card role.

QFX3500 and QFX3600 switches can be interconnected using VCPs to form a Virtual Chassis that is non-mixed; you do not need to configure mixed mode for a Virtual Chassis composed of only QFX3500 and QFX3600 switches. QFX3500 and QFX3600 switches can also be in a mixed QFX Series Virtual Chassis with EX4300 switches. In this combination, only QFX3500 or QFX3600 switches should be configured into the Routing Engine role, and the remaining members can be QFX3500, QFX3600, or EX4300 switches configured into line-card role.

EX4300 switches (excluding multigigabit models) can also be interconnected into a mixed Virtual Chassis with EX4600 switches. See the next section for information on mixed EX4300 and EX4600 Virtual Chassis.

## Understanding Mixed EX4300 and EX4600 Virtual Chassis

Any EX4300 switches (except for multigigabit models) and EX4600 switches can be interconnected into a Virtual Chassis.

In a mixed EX4300 and EX4600 Virtual Chassis:

- You can interconnect up to ten member switches.

- An EX4600 switch automatically assumes the primary Routing Engine role.
- EX4300 switches cannot assume the Routing Engine role.

EX4600 switches cannot be in a mixed Virtual Chassis with any other type of switch besides EX4300 switches that are not multigigabit model switches.

## Understanding EX4300 Multigigabit and Other EX4300 Model Switches in a Mixed EX4300 Virtual Chassis

You can combine EX4300 multigigabit model (EX4300-48MP) switches with other EX4300 switches into a mixed EX4300 Virtual Chassis.

**NOTE:** If an EX4300 Virtual Chassis has only EX4300 multigigabit model switches, or only a combination of other models of EX4300 switches, the Virtual Chassis is a non-mixed Virtual Chassis and you should not configure mixed mode.

In a mixed EX4300 Virtual Chassis:

- You can interconnect up to ten member switches.
- You must include the `ieee-clause-82` option when setting mixed mode on the EX4300 switches in the Virtual Chassis that are not multigigabit model switches. This option sets a special port mode (IEEE Clause 82) on the Virtual Chassis ports (VCPs) that enables them to communicate when interconnected with VCPs on EX4300 multigigabit switches. See *Configuring an EX2300, EX3400, or EX4300 Virtual Chassis* for details.
- Only EX4300 multigigabit model switches can be in the Routing Engine role.
- EX4300 switches that are not multigigabit model switches must be configured into line-card role.

EX4300 multigigabit switches cannot be in a mixed Virtual Chassis with any other type of switch besides other EX4300 switches.

**NOTE:** If you remove a non-multigigabit model EX4300 switch from a mixed EX4300 Virtual Chassis with multigigabit model members, you must disable `ieee-clause-82` port mode on the removed switch if you want to reconfigure it as a standalone switch or use it in any other type of mixed Virtual Chassis or any non-mixed Virtual Chassis. Otherwise, the VCPs will not connect with other members in the new Virtual Chassis. See *Removing or Replacing a Member Switch of a Virtual Chassis Configuration* for more details.

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

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

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

### RELATED DOCUMENTATION

*Configuring an EX2300, EX3400, or EX4300 Virtual Chassis*

*Virtual Chassis Overview for Switches*

*Understanding EX Series Virtual Chassis*

*Understanding QFX Series Virtual Chassis*

*Understanding Virtual Chassis Components*

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

All switches that are interconnected in a Virtual Chassis configuration are member switches of that Virtual Chassis. Each Virtual Chassis configuration has one member that functions as the *primary* in a Routing Engine role and controls the Virtual Chassis configuration. A second member functions as the backup in the Routing Engine role, and takes control if the primary fails.

When a Virtual Chassis configuration boots, it automatically runs a primary-role election algorithm to determine which member switch assumes the primary role. A Virtual Chassis also applies the same algorithm to choose a new primary or backup member if the member in either role fails.

The first factor the Virtual Chassis considers when choosing the primary member is the *primary-role priority*. On all members, the primary-role priority value is 128 by default. That value can change based on how you provision the Virtual Chassis, as follows:

- In a nonprovisioned Virtual Chassis, you can manually assign primary-role priority values from 0 to 255. A member with primary-role priority 0 will never be elected as primary (or backup), and always

stays in the linecard role. In this way, you configure higher primary-role priority values to specify which members can take on the primary (or backup) role.

- In a preprovisioned Virtual Chassis, you can't manually set the primary-role priority on any members. Instead, you assign the Routing Engine role to two member switches. The Virtual Chassis automatically changes the default primary-role priority (128) to 129 on those two members. Then the Virtual Chassis assigns the primary operate in the linecard role by default (you can also explicitly configure them with that role). The Virtual Chassis will never elect a linecard role member as a primary or backup member.

The primary-role election algorithm compares the Virtual Chassis members against the following criteria, in the order listed, until only one member remains under consideration. That member becomes the primary:

1. Choose the member with the highest primary-role priority.
2. Choose the member that was the primary the last time you rebooted the Virtual Chassis.
3. Choose the member that has been in the Virtual Chassis configuration for the longest period of time. (The member switches under consideration must have more than 1 minute between power-up times for this condition to make a difference.)
4. Choose the member with the lowest MAC address.

The primary-role election algorithm does not consider the different switch models or platforms in the Virtual Chassis. For some Virtual Chassis that can contain different types of switches, we require or recommend you configure certain switches in the primary and backup Routing Engine roles. See *Understanding Mixed EX Series and QFX Series Virtual Chassis* for details on the types of switches that can be mixed in a Virtual Chassis and which switches can or must be the primary or backup members.

To make sure a specific member is elected as the primary:

1. Power on only the switch that you want to be the primary in the Virtual Chassis.
2. (For a non-provisioned Virtual Chassis) Manually configure the primary-role priorities as follows:
  - a. Configure the highest possible primary-role priority value (255) on the member from the first step.
  - b. On the same member, which is now the primary, configure priority values on the other members. (For example, use the same value or next-highest value on the member you want to be the backup, and lower values on the other members.)
3. (For a preprovisioned Virtual Chassis) Configure the Routing Engine role on the two members that you want to act as the primary and backup members. (You can also explicitly configure the remaining members into the linecard role.)
4. Power on the other members.

You usually want to assign the same (highest) primary-role priority value to the members you want to be the primary and backup Routing Engine members to ensure reliable graceful Routing Engine switchover (GRES) operation. For either non-provisioned or preprovisioned Virtual Chassis, the other election considerations also help keep the primary role from switching back and forth rapidly between the two Routing Engine members under failover conditions.

For more information on configuring a Virtual Chassis, see ["Configuring an EX3300 Virtual Chassis \(CLI Procedure\)" on page 240](#), ["Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)" on page 245](#), [Configuring an EX2300, EX3400, or EX4300 Virtual Chassis](#), or [Configuring an EX4650 or a QFX Series Virtual Chassis](#).

## RELATED DOCUMENTATION

*Virtual Chassis Overview for Switches*

*Understanding EX Series Virtual Chassis*

*Understanding QFX Series Virtual Chassis*

*Understanding Virtual Chassis Components*

## Understanding Software Upgrades in a Virtual Chassis

### IN THIS SECTION

- [Automatic Software Updates | 50](#)
- [Nonstop Software Upgrade | 50](#)

This topic discusses software upgrades on EX Series and QFX Series *Virtual Chassis*, except EX8200 Virtual Chassis. For information on software upgrades on an EX8200 Virtual Chassis, see *Understanding Software Upgrades in an EX8200 Virtual Chassis*. For information on software upgrades on a Virtual Chassis Fabric (VCF), see *Understanding Software Upgrades in a Virtual Chassis Fabric*.

In a Virtual Chassis, each member switch must be running the same version of Juniper Networks Junos operating system (Junos OS) that supports Virtual Chassis. You can install a new Junos OS release on the entire Virtual Chassis or on individual members in the Virtual Chassis by using the same CLI command that you use to install Junos OS on standalone switches—the `request system software add` command.

In a mixed Virtual Chassis, the member switches must also be running the same version of Junos OS, but you might need to specify multiple Junos OS images when manually or automatically upgrading a mixed Virtual Chassis. For example, for the same Junos OS release, an EX4200 switch runs a different Junos OS image than an EX4500 or EX4550 switch, or an EX4300 switch runs a different Junos OS image than a QFX5100 switch in a QFX5100 Virtual Chassis. You can upgrade all member switches simultaneously by specifying a path to multiple Junos OS images in the same `request system software add` command.

**NOTE:** Some different switch platforms can run the same Junos OS image. For example, you do not need to use multiple Junos OS images when updating a Virtual Chassis with a combination of EX4500 and EX4550 switches, or a Virtual Chassis with a combination of QFX3500 and QFX3600 switches.

You can also use the following features to upgrade software on members of a Virtual Chassis:

## Automatic Software Updates

You can use the automatic software update feature on a non-mixed or mixed Virtual Chassis to automatically update the Junos OS version on member switches as you add them to the Virtual Chassis. See *Understanding Automatic Software Update on Virtual Chassis Member Switches* for more information.

If you are not configuring the automatic software update feature, we recommend that you update the new member switch to the version of Junos OS running on the Virtual Chassis before adding the member switch to the Virtual Chassis.

## Nonstop Software Upgrade

You can also use nonstop software upgrade (NSSU) to upgrade Junos OS on all members of a Virtual Chassis for EX Series and QFX Series Virtual Chassis that support NSSU. NSSU provides an orderly upgrade of each member of the Virtual Chassis and takes advantage of *graceful Routing Engine switchover*, *nonstop active routing*, and link aggregation to minimize traffic disruption during the upgrade.

For more information about NSSU, see:

- *Understanding Nonstop Software Upgrade on a Virtual Chassis and Mixed Virtual Chassis*
- *Upgrading Software on a Virtual Chassis and Mixed Virtual Chassis Using Nonstop Software Upgrade*
- (For legacy EX Series switches in a Virtual Chassis) *Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis (CLI Procedure)*

If NSSU is not supported to upgrade a Virtual Chassis from the currently-installed release to the release you need, refer to procedures like these for some guidance:

- [Two-Member QFX Series Virtual Chassis Upgrade Procedure](#), a network configuration example on how to manually upgrade a two-member QFX Series Virtual Chassis when NSSU isn't available.

## RELATED DOCUMENTATION

---

*Understanding Virtual Chassis Components*

---

*Configuring Automatic Software Update on Virtual Chassis Member Switches*

---

[Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine \(CLI Procedure\)](#)

---

[Installing Software on a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Switches \(CLI Procedure\) | 259](#)

## Understanding Global Management of a Virtual Chassis

### IN THIS SECTION

- [Console Port Session Redirection to Primary Switch | 52](#)
- [Logical Port for Virtual Chassis Out-of-Band Management | 53](#)

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

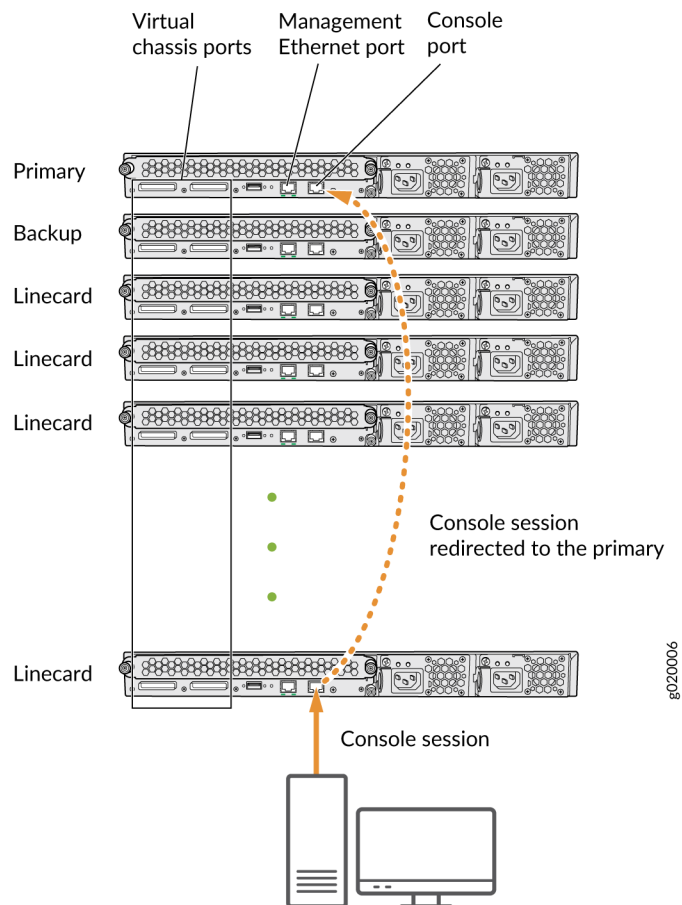
A Virtual Chassis is composed of multiple switches, so it has multiple console ports and multiple out-of-band management Ethernet ports located on the switches.



## Console Port Session Redirection to Primary Switch

You can connect a PC or laptop directly to a console port of any member switch to set up and configure the Virtual Chassis. When you connect to the console port of any member switch, the console session is redirected to the primary switch, as shown in [Figure 1 on page 52](#).

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



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

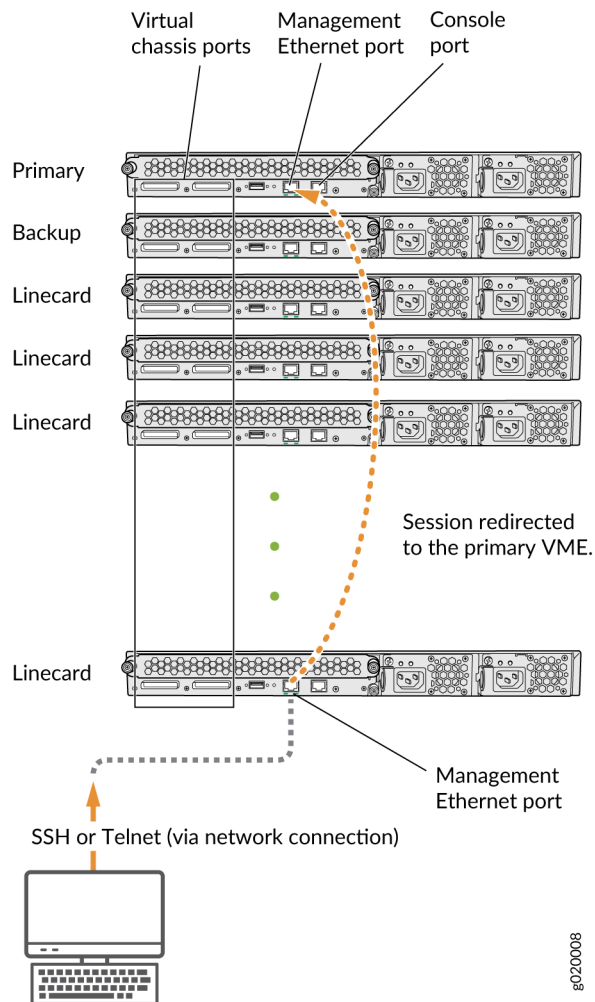
An out-of-band management Ethernet port is often referred to simply as a management Ethernet port. It uses a dedicated management channel for device maintenance and allows a system administrator to monitor and manage the switch by remote control.

## Logical Port for Virtual Chassis Out-of-Band Management

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

The Virtual Chassis configuration can be managed remotely through SSH or Telnet using a global management interface called the virtual management Ethernet (VME) interface. The VME interface is a *logical interface* representing all of the out-of-band management ports on the member switches. When you connect to the Virtual Chassis configuration using the VME interface's IP address, the connection is redirected to the primary member as shown in [Figure 2 on page 53](#).

**Figure 2: Management Ethernet Port Redirection to the VME Interface**



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

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

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

**NOTE:** In a QFX Series Virtual Chassis with QFX5110, QFX5120, or QFX5200 switches, the VME interface doesn't redirect properly to the management interface on the backup or linecard member switches if the management link on the primary member switch is down. Instead, you can use the console port on any member switch to globally configure or monitor the Virtual Chassis; this problem doesn't affect console port redirection (see "[Console Port Session Redirection to Primary Switch](#)" on page 52).

## RELATED DOCUMENTATION

*Understanding Virtual Chassis Components*

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

[Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\) | 295](#)

## Understanding the High-Speed Interconnection of the Dedicated Virtual Chassis Ports Connecting EX4200, EX4500, and EX4550 Member Switches

Two high-speed, dedicated *Virtual Chassis* ports (VCPs) on the EX4200 switches and on the Virtual Chassis modules on the EX4500 and EX4550 switches enable the member switches to be interconnected and operate as a single, powerful device. When dedicated VCP interfaces are used to form a ring topology, each segment provides 64 Gbps bidirectional bandwidth on an EX4200 switch and 128 Gbps bidirectional bandwidth on an EX4500 or EX4550 switch. Because the VCP links act as point-to-point links, multiple segments of the ring can be used simultaneously. This allows the Virtual Chassis configuration bandwidth to scale as you interconnect more members within the ring topology.

## RELATED DOCUMENTATION

*Understanding Virtual Chassis Components*

*Virtual Chassis Cabling Configuration Examples for EX4200 Switches*

## Understanding Virtual Chassis Port Link Aggregation

### IN THIS SECTION

- [Virtual Chassis Network Interface LAG Among Virtual Chassis Members | 55](#)
- [Virtual Chassis Port LAG Between Two Virtual Chassis Members | 56](#)

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

Within a Virtual Chassis, you can combine physical Ethernet ports to form a logical point-to-point link known as a *link aggregation group (LAG)* or *bundle*. The interfaces that are included in a LAG are sometimes referred to as member interfaces. Do not confuse this term with member switches, which refers to switches that are interconnected into a Virtual Chassis. A LAG in a Virtual Chassis can be composed of member interfaces located in different member switches in the Virtual Chassis, or composed of multiple redundant Virtual Chassis Port (VCP) links between two member switches in the Virtual Chassis, as described in the following sections.

### Virtual Chassis Network Interface LAG Among Virtual Chassis Members

When setting up interfaces in a Virtual Chassis, you can configure a combination of physical Ethernet ports belonging to different member switches to form a LAG. A LAG provides more bandwidth than a single Ethernet link can provide. Additionally, link aggregation provides network redundancy by load-balancing traffic across all available links. If one of the links fails, the system automatically load-balances traffic across all remaining links.

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

**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.

## Virtual Chassis Port LAG Between Two Virtual Chassis Members

You can configure optical uplink ports into Virtual Chassis ports (VCPs) that connect EX Series or QFX series switches together to form a Virtual Chassis. When you configure multiple uplink port VCPs connecting the same two member switches, those ports *automatically* form a LAG if the ports are configured to operate at the same link speeds. Each LAG is assigned a positive-integer identifier called a *trunk ID*. Up to 8 redundant VCP links can form a VCP LAG connecting two members in a Virtual Chassis, depending on the number of available ports that can be VCPs.

**NOTE:** On EX2200 and EX2200-C switches, you can also configure the RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T RJ-45 transceivers, into VCPs. Any interfaces on these switches that are configured as VCPs interconnecting two members will automatically form a LAG, regardless of whether the interfaces are optical transceiver interfaces, RJ-45 transceiver interfaces, or built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors.

A VCP LAG automatically forms when any two member switches are interconnected with two or more VCP links of the same link speed in any of these configurations:

- If the VCP ports on both switches are ports you configured into VCPs or default-configured VCPs (for switches that have default VCPs).
- If the VCP ports on both switches are dedicated VCPs (for switches that have dedicated VCPs).
- In a mixed Virtual Chassis when the VCP links interconnect two different switch models.

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

See *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port* or *Configuring an EX4650 or a QFX Series Virtual Chassis* for information about configuring uplink ports into VCPs.

## RELATED DOCUMENTATION

*Understanding EX Series Virtual Chassis*

[Understanding Aggregated Ethernet Interfaces and LACP for Switches](#)

[Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch | 191](#)

[Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch | 200](#)

[Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets | 148](#)

[Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports | 217](#)

*Understanding QFX Series Virtual Chassis*

*Configuring an EX4650 or a QFX Series Virtual Chassis*

## Understanding Fast Failover in an EX Series Virtual Chassis

### IN THIS SECTION

- [Supported Topologies for Fast Failover | 58](#)
- [How Fast Failover Works | 58](#)
- [Effects of Topology Changes on a Fast Failover Configuration | 65](#)

The *Virtual Chassis* fast failover feature is a hardware-assisted failover mechanism that automatically reroutes traffic and reduces traffic loss in the event of a link failure or switch failure in a Virtual Chassis. If a link between two members fails, traffic flow between those members must be rerouted quickly so that there is minimal traffic loss.

Fast failover is effective only for Virtual Chassis members configured in ring topologies using identical port types.

Fast failover is supported for the following Virtual Chassis:

- EX3300 Virtual Chassis
- EX4200 Virtual Chassis
- Mixed EX4200 and EX4500 Virtual Chassis

- Mixed EX4200 and EX4550 Virtual Chassis
- Mixed EX4200, EX4500, and EX4550 Virtual Chassis
- EX4500 Virtual Chassis
- Mixed EX4500 and EX4550 Virtual Chassis
- EX4550 Virtual Chassis

In any mixed Virtual Chassis, fast failover is supported for all Virtual Chassis ports (VCPs), including VCPs that provide interconnections between different types of member switches.

This topic describes the following:

## **Supported Topologies for Fast Failover**

For fast failover to be effective, the Virtual Chassis members must be configured in a ring topology. The ring topology can be formed by using either dedicated VCPs or user-configured uplink VCPs. Fast failover is supported only in a ring topology that uses identical port types, for example, either a topology that uses all dedicated VCPs or one that uses all uplink VCPs. Fast failover is not supported in a ring topology that includes both dedicated VCPs and uplink VCPs. Fast failover is supported, however, in a Virtual Chassis configuration that consists of multiple rings.

## **How Fast Failover Works**

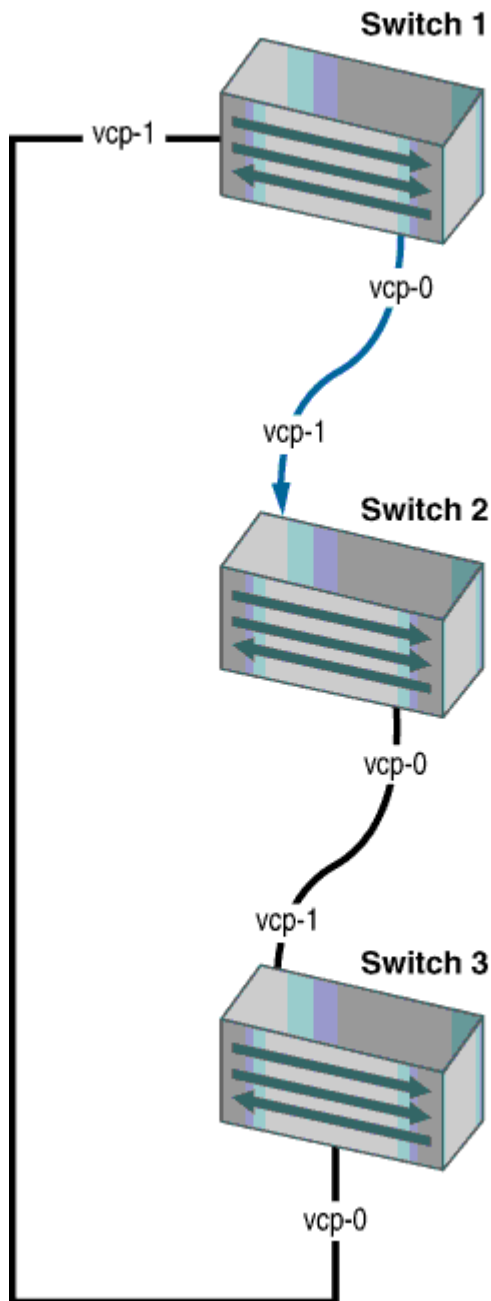
When fast failover is activated, each VCP is automatically configured with a backup port of the same type (dedicated VCP, SFP uplink VCP, or XFP uplink VCP). If a VCP fails, its backup port is used to send traffic. These backup ports act as standby ports and are not meant for load-balancing traffic or any other purposes.

### Fast Failover in a Ring Topology Using Dedicated VCPs

When fast failover is activated in a ring topology that uses dedicated VCPs, each VCP is automatically configured with a backup port of the same type. If a VCP fails, its backup port is used to send traffic.

[Figure 3 on page 59](#) shows normal traffic flow in a ring topology using dedicated VCPs.

**Figure 3: Normal Traffic Flow in a Ring Topology Using Dedicated VCPs**

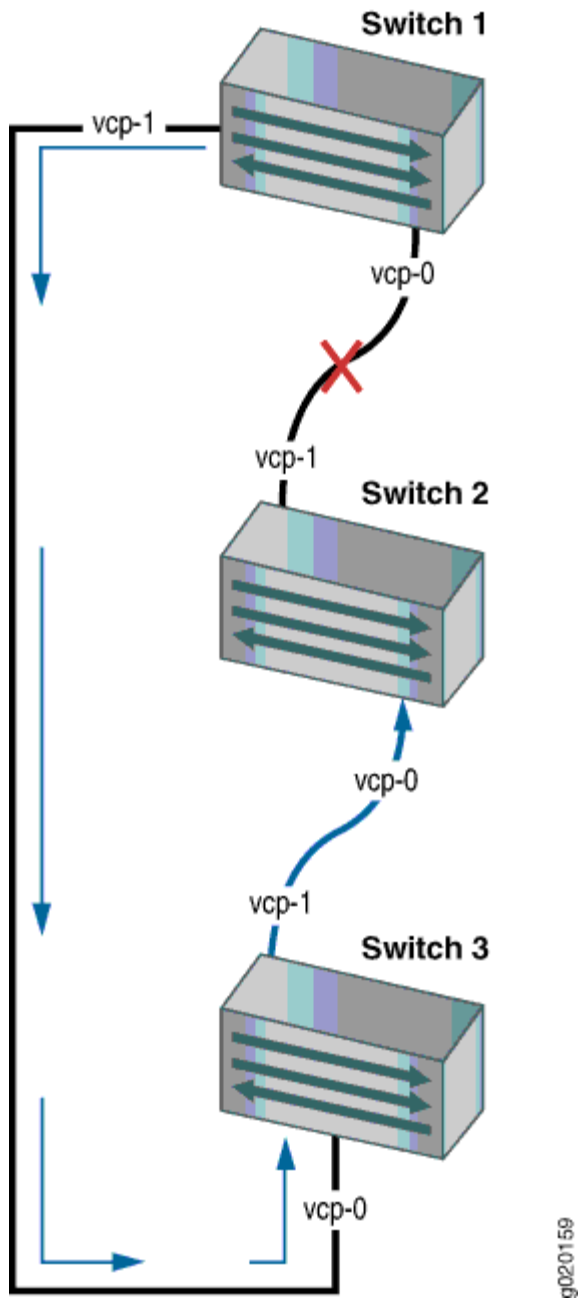


g020158



Figure 4 on page 60 shows traffic redirected by fast failover.

Figure 4: Traffic Redirected by Fast Failover After a Dedicated VCP Link Failure



When the failed link is restored, the Virtual Chassis reconfigures the topology to the topology's original state.

## Fast Failover in a Ring Topology Using Uplink Module VCPs

In a ring topology that uses uplink VCPs, each uplink VCP is automatically configured with a backup uplink VCP. If an uplink VCP fails, its backup port is used to send traffic. [Figure 5 on page 62](#) shows normal traffic flow in a ring topology using SFP uplink VCPs. Normal traffic flow in a ring topology using XFP uplink VCPs is the same.

**NOTE:** To use SFP or XFP uplink ports as VCPs, you must configure them to be VCPs using the **request virtual-chassis vc-port** command. Once configured, they are converted into VCPs. For example, **xe-0/1/0** becomes **vcp-255/1/0** after you configure it to be a VCP.

Figure 5: Normal Traffic Flow in a Ring Topology Using SFP Uplink VCPs

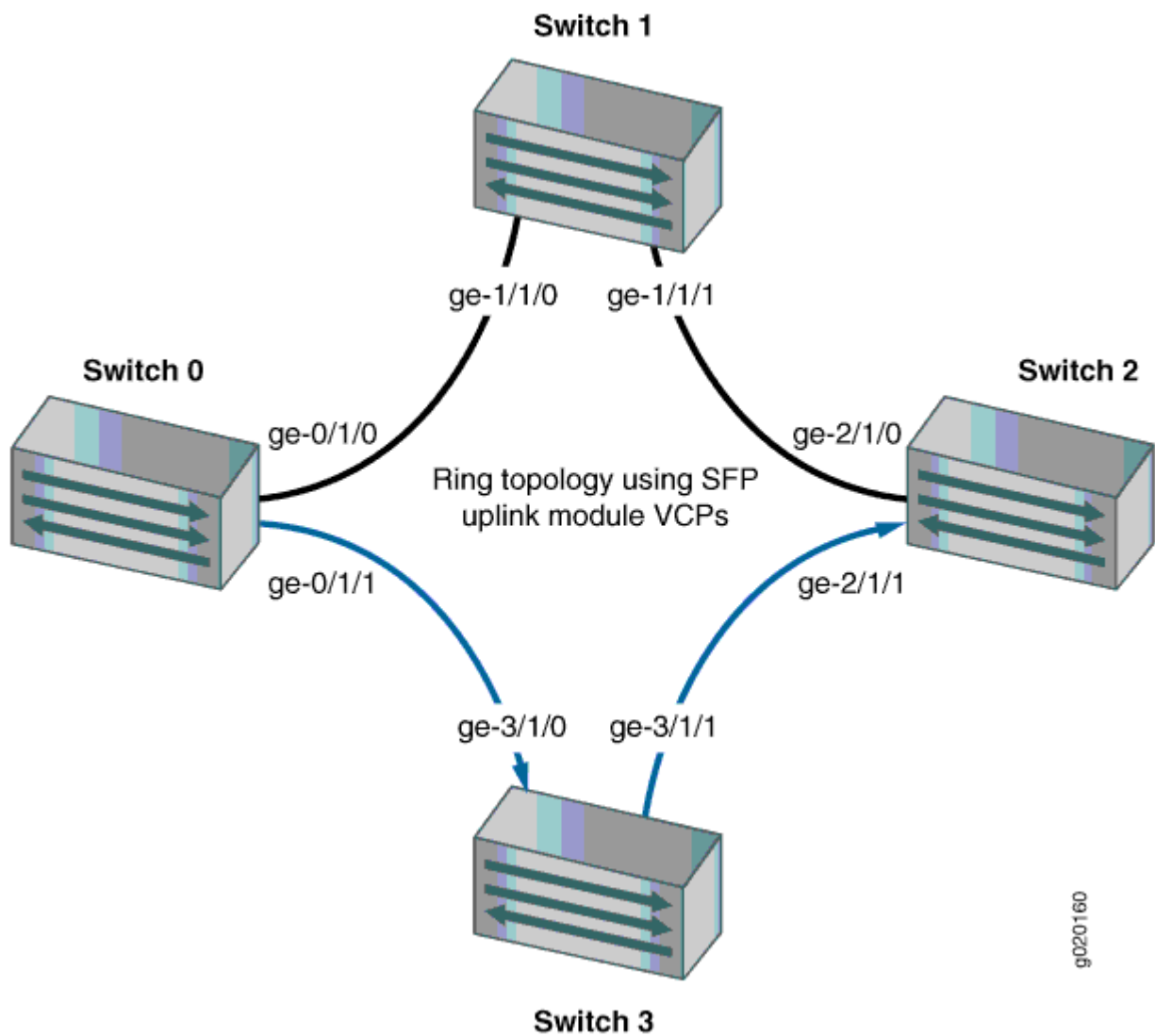
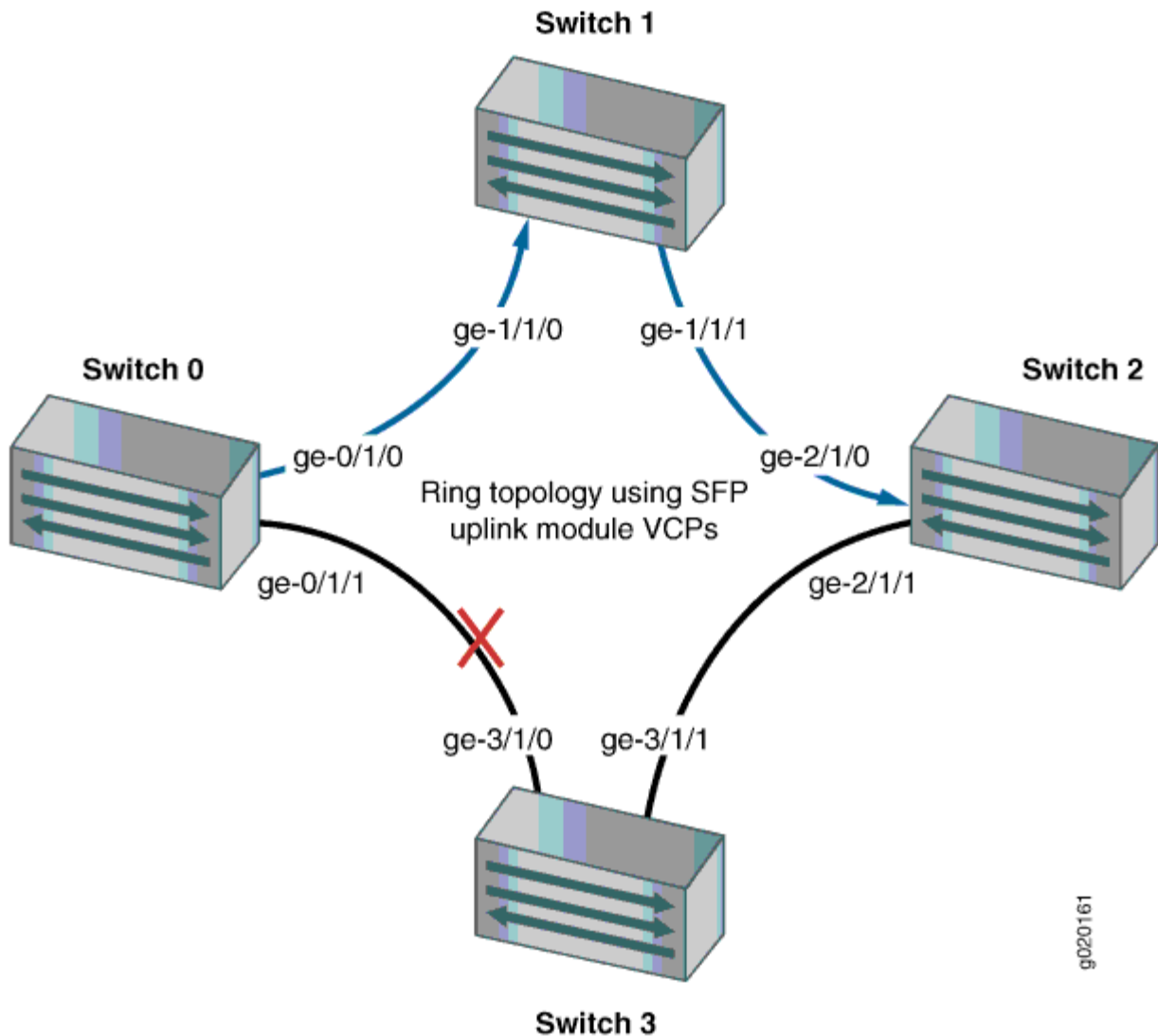


Figure 6 on page 63 shows traffic redirected by fast failover.

Figure 6: Traffic Redirected by Fast Failover After SFP Uplink VCP Link Failure



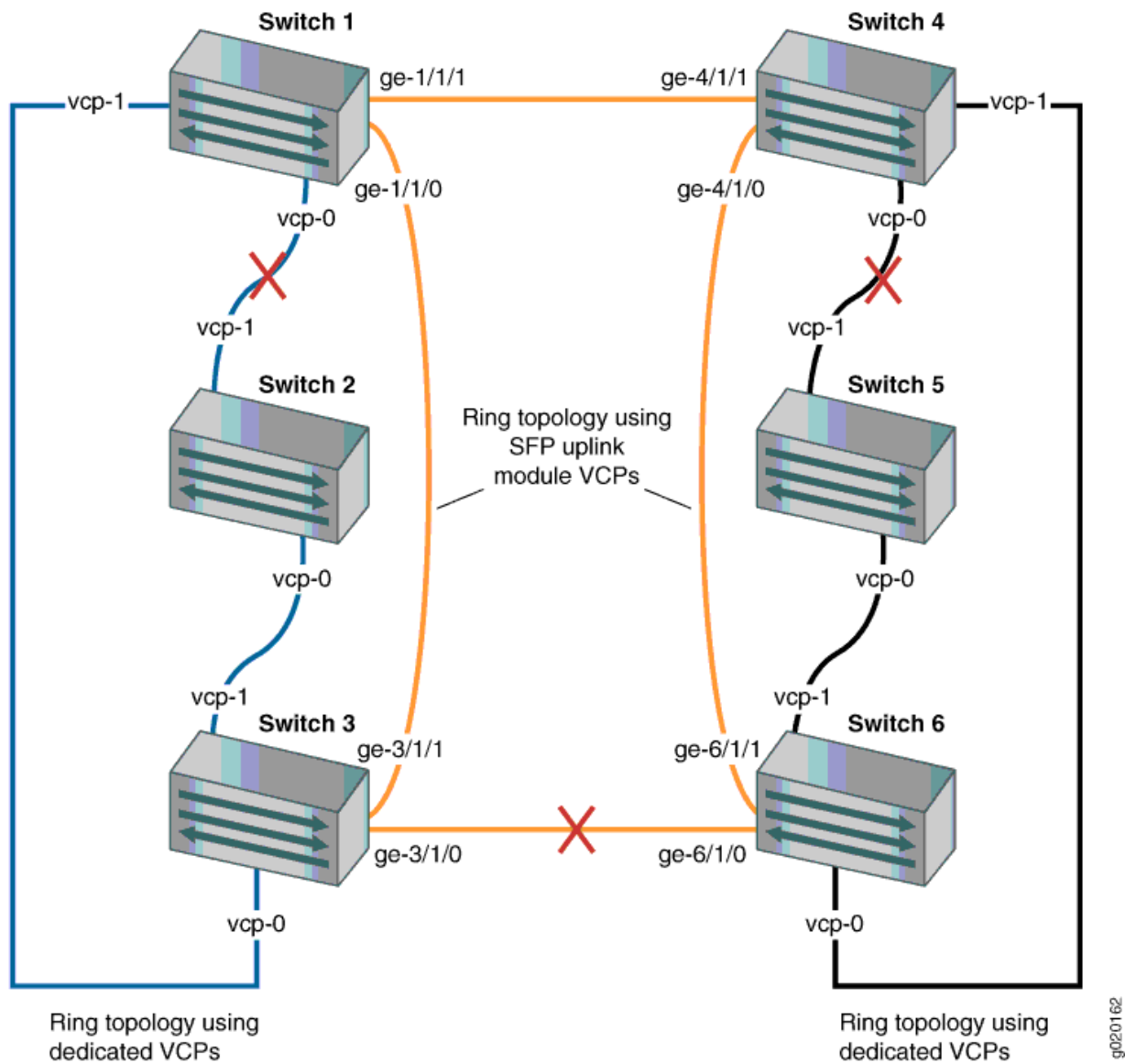
In a ring topology that uses SFP uplink VCPs, there are four ports per module. Consecutive pairs of ports are automatically configured as backup ports for each other. For example, if a Virtual Chassis member has an SFP uplink module installed, uplink module VCPs `ge-0/1/0` and `ge-0/1/1` are automatically configured as backup ports for each other. Similarly, ports `ge-0/1/2` and `ge-0/1/3` are automatically configured as backup ports for each other.

In a ring topology that uses XFP uplink module VCPs, there are only two ports per uplink module. Similarly to a topology that uses SFP uplink module VCPs, each port is automatically configured to back up the other port in the uplink module (for example, `xe-0/1/0` is the backup for `xe-0/1/1`).

## Fast Failover in a Virtual Chassis Configuration Using Multiple Ring Topologies

Fast failover is supported in a Virtual Chassis configuration with a multiple-ring topology, as shown in [Figure 7 on page 64](#).

**Figure 7: Traffic Redirected by Fast Failover After VCP Link Failures in a Topology with Multiple Rings**



In this scenario, the Virtual Chassis configuration has three rings: two rings that use dedicated VCPs and one ring that uses SFP uplink module VCPs. Fast failover works independently on each ring. Each dedicated VCP in a ring is backed up by another dedicated VCP. Similarly, each SFP uplink module VCP is backed up by another SFP uplink module VCP. Fast failover does not support a ring topology consisting of a mix of dedicated VCPs and uplink module VCPs.

## Effects of Topology Changes on a Fast Failover Configuration

When the fast failover feature is activated, topology changes to the Virtual Chassis configuration do not affect the fast failover configuration. In the event of a link or switch failure, fast failover functions normally.

### RELATED DOCUMENTATION

*Understanding EX Series Virtual Chassis*

[Understanding Fast Failover in an EX Series Virtual Chassis | 57](#)

*Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*

## Understanding Split and Merge in a Virtual Chassis

### IN THIS SECTION

- [What Happens When a Virtual Chassis Configuration Splits | 66](#)
- [Merging Virtual Chassis Configurations | 67](#)

In a Virtual Chassis, you connect two or more switches together to form a unit that is managed as a single chassis. If member switches in the Virtual Chassis fail or you remove member switches, this disrupts the Virtual Chassis configuration. In some situations, the Virtual Chassis configuration splits into two separate Virtual Chassis, which can cause disruptions in the network if the two resulting Virtual Chassis share common resources such as global IP addresses.

The Virtual Chassis split and merge feature is a method to prevent the separate Virtual Chassis configurations from adversely affecting the network. It also enables the two parts to merge back into a single Virtual Chassis configuration.

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

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

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

## What Happens When a Virtual Chassis Configuration Splits

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

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

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

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

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

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

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

## Merging Virtual Chassis Configurations

There are two scenarios in which separate Virtual Chassis merge:

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

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

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

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

When you connect two Virtual Chassis configurations:

1. Connecting the two split Virtual Chassis configurations triggers the shortest-path-first (SPF) algorithm. The SPF algorithm computes the network topology and then triggers the primary-role election algorithm. The primary-role election algorithm waits for the members to synchronize the topology information before running.
2. The primary-role election algorithm merges the VCIDs of all the members.
3. Each member runs the primary-role election algorithm to select a primary and a backup from among all members with the same VCIDs. For more information, see *Understanding How the Primary in a Virtual Chassis Is Elected*.
4. The primary determines whether the Virtual Chassis configuration is active or inactive. (See ["What Happens When a Virtual Chassis Configuration Splits" on page 66.](#))
5. If the Virtual Chassis configuration is active, the primary assigns roles to all members. If the Virtual Chassis configuration is inactive, the primary assigns all members the role of linecard.



6. When the other members receive their role from the primary, they change their role to backup or linecard. They also use the active or inactive state information sent by the primary to set their own state to active or inactive and to construct the Virtual Chassis member list from the information sent by the primary.
7. If the Virtual Chassis state is active, the primary waits for messages from the members indicating that they have changed their roles to the assigned roles, and then the primary changes its own role to primary.

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

## RELATED DOCUMENTATION

*Disabling Split and Merge in a Virtual Chassis*

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

[Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge | 213](#)

*Understanding EX Series Virtual Chassis*

*Understanding QFX Series Virtual Chassis*

## Understanding Automatic Software Update on Virtual Chassis Member Switches

### IN THIS SECTION

- [Automatic Software Update Basics | 69](#)
- [Automatic Software Update Restrictions | 69](#)

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

## Automatic Software Update Basics

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

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

## Automatic Software Update Restrictions

Refer to [Feature Explorer](#) to see the EX Series and QFX Series Virtual Chassis that support automatic software updates, and the Junos OS release versions where the feature was introduced on each platform. Where the automatic software update feature is supported, see the Junos OS Release Notes for the release version running on the primary for any limitations in using automatic software updates between that release version and any prospective member switch release versions.



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

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

## RELATED DOCUMENTATION

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

## Understanding Nonstop Software Upgrade on EX Series Switches

### SUMMARY

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

### IN THIS SECTION

- [Requirements for Performing an NSSU | 72](#)
- [How an NSSU Works | 73](#)
- [NSSU Limitations | 76](#)
- [NSSU and Junos OS Release Support | 77](#)
- [Overview of NSSU Configuration and Operation | 78](#)

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 7 on page 77 lists the EX Series switches and Virtual Chassis that support NSSU and the Junos OS release at which they began supporting it.

**Table 7: 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

**Table 7: Platform and Release Support for NSSU (Continued)**

Platform	Junos OS Release
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
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 72](#). 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 Device Reachability Testing

You can use a *Virtual Chassis* device reachability test to verify that the connections that connect the Virtual Chassis member devices together are functioning properly. You typically use a device reachability test to verify host reachability and connectivity within a Virtual Chassis.

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 provide information that is helpful in verifying connectivity between Virtual Chassis member devices.

### RELATED DOCUMENTATION

[Understanding Virtual Chassis Device Reachability Testing](#) | 79

## Understanding MAC Address Assignment on a Virtual Chassis

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

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

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

For Layer 2 and aggregated Ethernet interfaces, the Virtual Chassis assigns a unique MAC address that is derived from the member switch MAC address to each interface. The assignment of a unique MAC address to each network interface helps ensure that functions that require MAC address differentiation—such as redundant trunk groups (RTGs), Link Aggregation Control Protocol (LACP), and general monitoring functions—can function properly.

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

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

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

## RELATED DOCUMENTATION

[Understanding MAC Address Assignment on an EX Series Switch](#)

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

*Understanding EX Series Virtual Chassis*

*EX8200 Virtual Chassis Overview*

*Understanding QFX Series Virtual Chassis*

## Understanding High Availability on an EX Series Virtual Chassis

You increase your network's high availability (HA) when you interconnect a Juniper Networks EX Series Ethernet switch into a *Virtual Chassis*. A Virtual Chassis is more fault tolerant than a standalone EX

series switch because it remains up when a single member switch fails, and provides sub-second convergence in the case of a device or link failure.

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

A Virtual Chassis has dual Routing Engines—the switch in the primary role and the switch in the backup role—and therefore supports many HA features not supported on standalone EX Series switches, such as Graceful Routing Engine Switchover (GRES) for hitless failover. For information on which of the High Availability features listed in [Table 8 on page 81](#) are supported in your EX Series Virtual Chassis, see [Feature Explorer](#).

Many HA features for the EX Series Virtual Chassis are designed to improve network resiliency after a Routing Engine switchover. [Table 8 on page 81](#) describes the effects of a Routing Engine switchover when no high availability features are enabled and when some High Availability features are enabled.

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

High Availability Feature	Effect of Routing Engine Switchover
No HA features enabled	Kernel and forwarding state information is not preserved to the backup Routing Engine. A convergence process that requires all interfaces on the Virtual Chassis to be taken offline has to be performed before the Virtual Chassis returns online. The switchover can take several minutes and the Virtual Chassis does not send or receive traffic until the switchover is complete.
<i>Graceful Routing Engine switchover</i> (GRES) enabled	Kernel and forwarding state information is preserved on both Routing Engines, so the convergence process does not occur and the switchover happens quickly with minimal traffic loss.

Table 8: Effects of a Routing Engine Switchover (*Continued*)

High Availability Feature	Effect of Routing Engine Switchover
<i>Nonstop active routing</i> (NSR), Nonstop bridging (NSB), or both enabled	<p>Layer 2 protocols that are supported by NSB are not disrupted by a Routing Engine switchover when NSB is enabled. Layer 2 protocol information for all active Layer 2 protocols is stored on both Routing Engines when NSB is enabled.</p> <p>Layer 3 protocols that are supported by NSR are not disrupted by a Routing Engine switchover when NSR is enabled. Layer 3 protocol information for all active Layer protocols is stored on both Routing Engines when NSR is enabled.</p>
Graceful Protocol Restart enabled	<p>Traffic is not interrupted during the switchover. Interface and kernel information is preserved. Graceful restart protocol extensions quickly collect and restore routing information for supported protocols from the neighboring devices.</p>

## RELATED DOCUMENTATION

*Understanding EX Series Virtual Chassis*

# 2

PART

## Configuration

---

[Configuration Examples](#) | 84

[Configuration Tasks](#) | 234

[Configuration Statements](#) | 312

---



# Configuration Examples

## IN THIS CHAPTER

- [Example: Configuring an EX2200 Virtual Chassis with Four Member Switches | 85](#)
- [Example: Configuring an EX3300 Virtual Chassis with a Primary and Backup | 91](#)
- [Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)
- [Example: Configuring an EX4500 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 105](#)
- [Example: Expanding an EX3300 Virtual Chassis | 112](#)
- [Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet | 119](#)
- [Example: Adding EX4500 Switches to a Preprovisioned Virtual Chassis | 128](#)
- [Example: Adding an EX4500 Switch to a Nonprovisioned Virtual Chassis | 134](#)
- [Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration | 140](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets | 148](#)
- [Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets | 159](#)
- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File | 168](#)
- [Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis | 183](#)
- [Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch | 191](#)
- [Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch | 200](#)
- [Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails | 208](#)
- [Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge | 213](#)
- [Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports | 217](#)
- [Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches | 229](#)

## Example: Configuring an EX2200 Virtual Chassis with Four Member Switches

### IN THIS SECTION

- [Requirements | 85](#)
- [Overview and Topology | 86](#)
- [Configuration | 87](#)
- [Verification | 89](#)
- [Troubleshooting the Virtual Chassis | 90](#)

A Virtual Chassis configuration is a scalable switch. You can provide secure, redundant network accessibility with an EX2200 Virtual Chassis. You can expand the Virtual Chassis configuration to include up to four EX2200 member switches to provide additional access interfaces as your office grows.

This example describes how to configure a four-member EX2200 Virtual Chassis. You can use this configuration with the same hardware and software configurations if the member switches are installed in the same or different wiring closets.

### Requirements

This example uses the following hardware and software components:

- Four EX2200 switches
- Four Virtual Chassis port (VCP)-configurable optical modules.

To confirm which optical modules are supported for EX2200 switches and are VCP-configurable, see [Pluggable Transceivers Supported on EX2200 Switches](#).

- Junos OS Release 12.2 or later for EX Series switches

Before you begin, be sure you have:

1. Rack-mounted the switches. See [Mounting an EX2200 Switch](#).
2. Cabled the switches. See [Installing and Connecting an EX2200 Switch](#).

## Overview and Topology

### IN THIS SECTION

- [Topology | 86](#)

This example shows a Virtual Chassis configuration composed of four EX2200 switches. You would typically use a four-member EX2200 Virtual Chassis to add ports at the access layer in a manner that minimized network complexity, or to combine two EX2200 access layer switches at different sites into a single switch. A Virtual Chassis reduces network complexity by allowing a single configuration to be applied to all member switches and by minimizing the need for a loop prevention protocol like Spanning Tree Protocol (STP).

You interconnect EX2200 switches into a Virtual Chassis by configuring uplink ports connecting EX2200 member switches as Virtual Chassis ports (VCPs). You must use the uplink ports configured as VCPs regardless of whether the member switches are installed in the same or different wiring closets.

After you interconnect the switches with the uplink ports and configure them as VCPs, the VCPs are operational. The user assigns the primary role priorities and member IDs using the Junos OS software in this example. The software elects a primary based on several criteria, including how long a member switch has belonged to the Virtual Chassis configuration. For additional details on how the primary switch is elected, see *Understanding How the Primary in a Virtual Chassis Is Elected*.

**NOTE:** This example uses a nonprovisioned configuration. You can explicitly assign a role for each member switch by preprovisioning the Virtual Chassis. See "[Configuring an EX2200 Virtual Chassis \(CLI Procedure\)](#)" on page 235.

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis.

### Topology

The topology for this example consists of four EX2200 switches.

[Table 9 on page 87](#) shows the default configuration settings for the Virtual Chassis.

Table 9: Components of the Basic Virtual Chassis Access Switch Topology

Member Switch	Hardware	Member ID	Role and Priority
switch-0	EX2200 switch	0	Primary: primary-role priority 255
switch-1	EX2200 switch	1	Backup: primary-role priority 255
switch-2	EX2200 switch	2	Line card; primary-role priority 128
switch-3	EX2200 switch	3	Line card; primary-role priority 128

## Configuration

### IN THIS SECTION

- [Procedure | 87](#)

## Procedure

### Step-by-Step Procedure

To configure a Virtual Chassis with four member switches in a single wiring closet:

1. Make sure the optical ports on the member switches are properly cabled. See *Install a Transceiver*.
2. Power on switch-0 (the member switch that you want to function as the primary).
3. When the CLI prompt becomes available, run the EZSetup program on switch-0, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* or *Connecting and Configuring an EX Series Switch (J-Web Procedure)* for details.

4. Configure switch-0 with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis configuration, if desired.

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

5. Configure the primary-role priority of the switches that you want to act in the primary and backup roles:

```
[edit virtual-chassis]
user@switch-0# set member 0 mastership-priority 255
user@switch-0# set member 1 mastership-priority 255
```

6. Power on switch-1. The default primary-role priority election algorithm uses the amount of time powered on to assign the backup role to a switch, so powering on switch-1 before switch-2 and switch-3 helps ensure the correct roles are assigned to each member switch.
7. Power on switch-2 and switch-3. You do not need to configure the primary-role priority values for these switches since they will use the default priority values of 128.
8. Configure the optical ports on each switch as VCPs.

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

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

```
user@switch-2> request virtual-chassis vc-port set pic-slot 1 port 0
user@switch-2> request virtual-chassis vc-port set pic-slot 1 port 1
```

```
user@switch-3> request virtual-chassis vc-port set pic-slot 1 port 0
user@switch-3> request virtual-chassis vc-port set pic-slot 1 port 1
```

## Verification

### IN THIS SECTION

- [Verifying That the Member Switches are Present and the Roles are Properly Assigned | 89](#)

To confirm that the Virtual Chassis is operational, perform these tasks:

### Verifying That the Member Switches are Present and the Roles are Properly Assigned

#### Purpose

Verify that the member switches are present in the Virtual Chassis and that the primary switch, which has been selected by default, is the member switch that you want to function in that role.

#### Action

List the member switches of the Virtual Chassis configuration:

```
user@switch-0> show virtual-chassis
```

```
Virtual Chassis ID: 5dfe.e61f.2abc
```

```
Virtual Chassis Mode: Enabled
```

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Neighbor List Mode ID	Interface
0 (FPC 0)	Prsnt	GR0211195292	ex2200-24t-4g	255	Master*	NA 1	vcp-255/1/0
						3	vcp-255/1/1
1 (FPC 1)	Prsnt	GP0211420824	ex2200-24t-4g	255	Backup	NA 0	vcp-255/1/0
						2	vcp-255/1/1
2 (FPC 2)	Prsnt	GP0211464311	ex2200-24t-4g	128	Linecard	NA 1	vcp-255/1/0
						3	vcp-255/1/1
3 (FPC 3)	Prsnt	CW0210108833	ex2200-24t-4g	128	Linecard	NA 2	vcp-255/1/0
						0	vcp-255/1/1

```
Member ID for next new member: None
```

## Meaning

The `show virtual-chassis` command lists the member switches interconnected in a Virtual Chassis configuration with the member IDs that have been assigned by the primary, the status of each member switch, the primary-role priority values, and the roles. It also displays the neighbor members with which each member is interconnected. The output shows that all switches are in the desired `Prsnt` state and that switch-0 and switch-1 are in the primary and backup roles.

## Troubleshooting the Virtual Chassis

### IN THIS SECTION

- [Troubleshooting the Assignment of Roles | 90](#)
- [Troubleshooting the VCPs | 91](#)

To troubleshoot the configuration of an EX2200 Virtual Chassis, perform these tasks:

### Troubleshooting the Assignment of Roles

#### Problem

The primary and backup roles are not assigned to the member switches that you want to function in these roles.

#### Solution

Modify the primary-role priority values.

To quickly modify the primary-role priority of any switch in the Virtual Chassis (in this case, member ID 1), enter the following command:

```
[edit virtual-chassis]
user@switch-0# set member 1 mastership-priority 255
```

## Troubleshooting the VCPs

### Problem

The VCPs are down.

### Solution

1. Check to make sure that you have cabled the appropriate ports.
2. Check to make sure that the cables are seated properly.
3. Check the output of the `show virtual-chassis vc-port` command to ensure you properly configured the VCPs. Enter the `request virtual-chassis vc-port` command to reconfigure the VCPs, if needed.

### RELATED DOCUMENTATION

| [Configuring an EX2200 Virtual Chassis \(CLI Procedure\) | 235](#)

## Example: Configuring an EX3300 Virtual Chassis with a Primary and Backup

### IN THIS SECTION

- [Requirements | 92](#)
- [Overview and Topology | 92](#)
- [Configuration | 93](#)
- [Verification | 94](#)
- [Troubleshooting the Virtual Chassis | 96](#)

A Virtual Chassis configuration is a scalable switch. You can provide secure, redundant network accessibility with a basic two-member EX3300 Virtual Chassis and later expand the Virtual Chassis configuration to provide additional access ports as your office grows.



This example describes how to configure an EX3300 Virtual Chassis with a primary and backup in a single wiring closet. You could use the same software configuration, however, if the EX3300 switches were connected across wiring closets.

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.3 or later for EX Series switches
- Two EX3300 switches

Before you begin, be sure you have:

1. Rack-mounted the switches. See [Mounting an EX3300 Switch](#).
2. Cabled the switches. See [Virtual Chassis Cabling Configuration Examples for EX3300 Switches](#).

## Overview and Topology

### IN THIS SECTION

- [Topology](#) | 93

A Virtual Chassis configuration allows you to accommodate the networking needs of a growing office. The default configuration of a two-member Virtual Chassis includes a primary and a backup switch. In addition to providing more access ports than a single switch can provide, a Virtual Chassis configuration provides high availability through redundancy.

This example shows a Virtual Chassis configuration composed of two EX3300 switches. By default, EX3300 switches have two uplink ports configured as Virtual Chassis ports (VCPs). The EX3300 Virtual Chassis is connected using these VCPs.

After you interconnect the switches with the VCPs and power on the switches, the VCPs are operational. The primary role priorities and member IDs are assigned by the software. The software elects a primary based on several criteria, including how long a member switch has belonged to the Virtual Chassis configuration. For additional details, see *Understanding How the Primary in a Virtual Chassis Is Elected*. Therefore, we recommend that you start by powering on only one member switch, the one that you want to function as the primary.

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis.

Topology

The topology for this example consists of two EX3300 switches.

[Table 10 on page 93](#) shows the default configuration settings for the two-member Virtual Chassis.

**Table 10: Components of the Basic Virtual Chassis Access Switch Topology**

Member Switch	Hardware	Member ID	Role and Priority
SWA-0	EX3300 switch	0	Primary: primary-role priority 128
SWA-1	EX3300 switch	1	Backup: primary-role priority 128

Configuration

IN THIS SECTION

- [Procedure | 93](#)

Configure a Virtual Chassis with a default primary and backup in a single wiring closet:

Procedure

Step-by-Step Procedure

To configure a Virtual Chassis with primary and backup:

1. Make sure the VCPs on the rear panel of the member switches are properly cabled. See [Virtual Chassis Cabling Configuration Examples for EX3300 Switches](#).

2. Power on SWA-0 (the member switch that you want to function as the primary).
3. Check the front-panel LCD to confirm that the switch has powered on correctly.
4. Run the EZSetup program on SWA-0, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* or *Connecting and Configuring an EX Series Switch (J-Web Procedure)* for details.
5. Configure SWA-0 with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis configuration, if desired.

```
[edit]
user@SWA-0# set interfaces vme unit 0 family inet address /ip-address/mask/
```

6. (Optional, but recommended) Disable the split and merge feature:

```
[edit virtual-chassis]
user@SWA-0# set no-split-detection
```

7. Power on SWA-1.

## Verification

### IN THIS SECTION

- [Verifying That the Primary-role Priority Is Assigned Appropriately | 94](#)
- [Verifying That the VCPs Are Operational | 95](#)

To confirm that the Virtual Chassis configuration is operational, perform these tasks:

### Verifying That the Primary-role Priority Is Assigned Appropriately

#### Purpose

Verify that the primary, which has been selected by default, is the member switch that you want to function in that role.

## Action

1. Check the front-panel LCD to confirm that the switch has powered on correctly and that a member ID has been assigned.
2. List the member switches of the Virtual Chassis configuration.

```
user@SWA-0> show virtual-chassis
```

```
Virtual Chassis ID: 0019.e250.47a0
```

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID Interface
0 (FPC 0)	Prsnt	AK0207360276	ex3300-24t	128	Master*	1 vcp-255/1/2 1 vcp-255/1/3
1 (FPC 1)	Prsnt	AK0207360281	ex3300-24t	128	Backup	0 vcp-255/1/2 0 vcp-255/1/3

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

## Meaning

The *show virtual-chassis* command lists the member switches interconnected in a Virtual Chassis configuration with the member IDs that have been assigned by the primary, the primary-role priority values, and the roles. It also displays the neighbor members with which each member is interconnected. The output shows that SWA-0, member 0, has been assigned default primary-role priority 128. Because SWA-0 is the first member to be powered on, it has the most seniority and is therefore assigned the role of primary. SWA-1 is powered on after member 0, so it is assigned the role of backup. The member IDs are displayed on the front panel of the switches. Check and confirm whether the default assignment is satisfactory.

## Verifying That the VCPs Are Operational

### Purpose

Verify that the VCPs interconnecting the switches are operational.

## Action

Display the VCPs of all the members:

```
user@SWA-0> show virtual-chassis vc-port all-members
fpc0:
-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
1/2             Configured Up       10000      1    vcp-255/1/2
1/3             Configured Up       10000      1    vcp-255/1/3

fpc1:
-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
1/2             Configured Up       10000      0    vcp-255/1/2
1/3             Configured Up       10000      0    vcp-255/1/3
```

## Meaning

The *show virtual-chassis vc-port* command lists the interfaces that are enabled for the member switches of the Virtual Chassis configuration and shows the status of the interfaces. The output in this example shows that two of the VCPs are operational and two VCPs are not. A single cable has been used to interconnect vcp-0 of member ID 0 and vcp-0 of member ID 1. That interconnection is sufficient for the switch to be operational. However, we recommend that you connect the second set of VCPs for redundancy.

## Troubleshooting the Virtual Chassis

### IN THIS SECTION

- [Troubleshooting the Assignment of Roles | 97](#)
- [Troubleshooting the VCPs | 97](#)

To troubleshoot the configuration of a Virtual Chassis, perform these tasks:

### Troubleshooting the Assignment of Roles

#### Problem

The primary and backup roles are not assigned to the member switches that you want to function in these roles.

#### Solution

Modify the primary-role priority values.

To quickly modify the primary-role priority of SWA-1 (member ID 1), copy the following command and paste it into the switch terminal window:

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

### Troubleshooting the VCPs

#### Problem

The VCPs are down.

#### Solution

1. Check to make sure that you have cabled the appropriate ports.
2. Check to make sure that the cables are seated properly.

You should generally cable and interconnect both of the VCPs on the member switches, for redundancy and high availability.

### RELATED DOCUMENTATION

| [Configuring an EX3300 Virtual Chassis \(CLI Procedure\)](#) | 240

## Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet

### IN THIS SECTION

- Requirements | 98
- Overview and Topology | 99
- Configuration | 100
- Verification | 101
- Troubleshooting the Virtual Chassis | 104

A Virtual Chassis configuration is a scalable switch. You can provide secure, redundant network accessibility with a basic two-member EX4200 Virtual Chassis and later expand the Virtual Chassis configuration to provide additional access ports as your office grows.

This example describes how to configure an EX4200 Virtual Chassis with a primary and backup in a single wiring closet:

### Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.0 or later for EX Series switches
- One EX4200-48P switch
- One EX4200-24T switch
- One XFP uplink module

Before you begin, be sure you have:

1. Rack-mounted the switches.
2. Installed the uplink module. See *Installing an Uplink Module in an EX4200 Switch*.
3. Cabled the switches. See *Connecting a Virtual Chassis Cable to an EX4200 Switch*.

## Overview and Topology

### IN THIS SECTION

- [Topology](#) | 99

A Virtual Chassis configuration allows you to accommodate the networking needs of a growing office. The default configuration of a two-member Virtual Chassis includes a primary and a backup switch. In addition to providing more access ports than a single switch can provide, a Virtual Chassis configuration provides high availability through redundancy.

This example shows a Virtual Chassis configuration composed of two EX4200 switches. One of the switches has an uplink module with ports that can be configured to connect to a distribution switch or customer edge (CE) router or that can be configured as Virtual Chassis ports (VCPs) to interconnect with a member switch that is located too far for the dedicated VCP cabling. (The network interfaces on EX4200-24F switches can also be configured as VCPs.) For information on configuring the uplink ports as trunk ports to a distribution switch, see [Configuring Gigabit Ethernet Interfaces \(CLI Procedure\)](#). For an example of configuring uplink ports as VCPs, see *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*.

By default, after you interconnect the switches with the dedicated VCPs and power on the switches, the VCPs are operational. The primary role priorities and member IDs are assigned by the software. The software elects a primary based on several criteria, including how long a member switch has belonged to the Virtual Chassis configuration. For additional details, see *Understanding How the Primary in a Virtual Chassis Is Elected*. Therefore, we recommend that you start by powering on only one member switch, the one that you want to function as the primary.

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis.

The Virtual Chassis configuration provides networking access for 50 onsite workers, who are sitting within range of a single wiring closet. The workers all use personal computers and VoIP phones. As the office grows, you can add more EX4200 switches to meet increased needs for access ports.

### Topology

The topology for this example consists of two switches, one of which contains an uplink module:

- One EX4200-48P switch (SWA-0) with 48 access ports, all of which support PoE



- One EX4200-24T switch (SWA-1) with 24 access ports, including eight ports that support PoE
- One XFP uplink module, with two 10-Gigabit Ethernet ports, is installed in the EX4200-48P switch

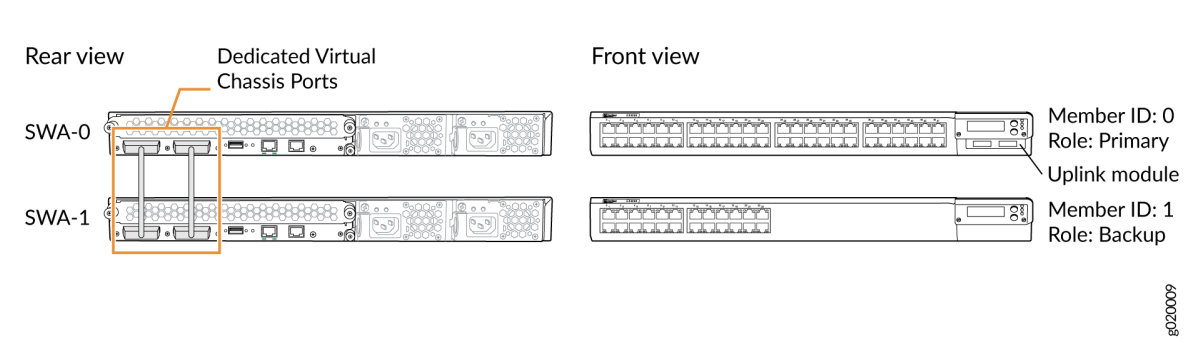
Table 11 on page 100 shows the default configuration settings for the two-member Virtual Chassis.

Table 11: Components of the Basic Virtual Chassis Access Switch Topology

Member Switch	Hardware	Member ID	Role and Priority
SWA-0	EX4200-48P switch	0	Primary: primary-role priority 128
SWA-1	EX4200-24T switch	1	Backup: primary-role priority 128

Figure 8 on page 100 shows that SWA-0 and SWA-1 are interconnected with their dedicated VCPs on the rear panel. The LCD on the front displays the member ID and role. SWA-0 also includes an uplink module. Its uplink ports can be used to connect to a distribution switch.

Figure 8: Basic EX4200 Virtual Chassis with Primary and Backup



## Configuration

### IN THIS SECTION

- Procedure | 101

Configure a Virtual Chassis with a default primary and backup in a single wiring closet:

## Procedure

### Step-by-Step Procedure

To configure a Virtual Chassis with primary and backup:

1. Make sure the VCPs on the rear panel of the member switches are properly cabled. See *Virtual Chassis Cabling Configuration Examples for EX4200 Switches*.
2. Power on SWA-0 (the member switch that you want to function as the primary).
3. Check the front-panel LCD to confirm that the switch has powered on correctly.
4. Run the EZSetup program on SWA-0, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* or *Connecting and Configuring an EX Series Switch (J-Web Procedure)* for details.
5. Configure SWA-0 with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis configuration, if desired.

```
[edit]
user@SWA-0# set interfaces vme unit 0 family inet address /ip-address/mask/
```

6. (Optional, but recommended) Disable the split and merge feature:

```
[edit virtual-chassis]
user@SWA-0# set no-split-detection
```

7. Power on SWA-1.

## Verification

### IN THIS SECTION

- [Verifying That the Primary-role Priority Is Assigned Appropriately | 102](#)
- [Verifying That the VCPs Are Operational | 103](#)

To confirm that the Virtual Chassis configuration is operational, perform these tasks:

## Verifying That the Primary-role Priority Is Assigned Appropriately

### Purpose

Verify that the primary, which has been selected by default, is the member switch that you want to function in that role.

### Action

1. Check the front-panel LCD to confirm that the switch has powered on correctly and that a member ID has been assigned.
2. List the member switches of the Virtual Chassis configuration.

```
user@SWA-0> show virtual-chassis
```

```
Virtual Chassis ID: 0019.e250.47a0
```

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID Interface
0 (FPC 0)	Prsnt	AK0207360276	ex4200-48p	128	Master*	1 vcp-0 1 vcp-1
1 (FPC 1)	Prsnt	AK0207360281	ex4200-24t	128	Backup	0 vcp-0 0 vcp-1

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

### Meaning

The *show virtual-chassis* command lists the member switches interconnected in a Virtual Chassis configuration with the member IDs that have been assigned by the primary, the primary-role priority values, and the roles. It also displays the neighbor members with which each member is interconnected. The output shows that SWA-0, member 0, has been assigned default primary-role priority 128. Because SWA-0 is the first member to be powered on, it has the most seniority and is therefore assigned the role of primary. SWA-1 is powered on after member 0, so it is assigned the role of backup. The member IDs are displayed on the front panel of the switches. Check and confirm whether the default assignment is satisfactory.

## Verifying That the VCPs Are Operational

### Purpose

Verify that the dedicated VCPs interconnecting the switches are operational.

### Action

Display the VCPs of all the members:

```

user@SWA-0> show virtual-chassis vc-port all-members
fpc0:
-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up       32000      1    vcp-1
vcp-1          Dedicated Up       32000      1    vcp-0

fpc1:
-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up       32000      1    vcp-1
vcp-1          Dedicated Up       32000      1    vcp-0

```

### Meaning

The *show virtual-chassis vc-port* command lists the interfaces that are enabled for the member switches of the Virtual Chassis configuration and shows the status of the interfaces. The output in this example shows that two of the VCPs are operational and two VCPs are not. A single cable has been used to interconnect vcp-0 of member ID 0 and vcp-0 of member ID 1. That interconnection is sufficient for the switch to be operational. However, we recommend that you connect the second set of VCPs for redundancy.

## Troubleshooting the Virtual Chassis

### IN THIS SECTION

- [Troubleshooting the Assignment of Roles | 104](#)
- [Troubleshooting the VCPs | 104](#)

To troubleshoot the configuration of a Virtual Chassis, perform these tasks:

### Troubleshooting the Assignment of Roles

#### Problem

The primary and backup roles are not assigned to the member switches that you want to function in these roles.

#### Solution

Modify the primary-role priority values.

To quickly modify the primary-role priority of SWA-1 (member ID 1), copy the following command and paste it into the switch terminal window:

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

### Troubleshooting the VCPs

#### Problem

The VCPs are down.

#### Solution

1. Check to make sure that you have cabled the appropriate ports.
2. Check to make sure that the cables are seated properly.

You should generally cable and interconnect both of the VCPs on the member switches, for redundancy and high availability.

## RELATED DOCUMENTATION

[Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet | 119](#)

[Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration | 140](#)

[Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File | 168](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

[Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)

## Example: Configuring an EX4500 Virtual Chassis with a Primary and Backup in a Single Wiring Closet

### IN THIS SECTION

- [Requirements | 105](#)
- [Overview and Topology | 106](#)
- [Configuration | 107](#)
- [Verification | 108](#)
- [Troubleshooting the Virtual Chassis | 111](#)

A Virtual Chassis configuration is a scalable switch. You can provide secure, redundant network accessibility with a basic two-member EX4500 Virtual Chassis configuration and later expand the Virtual Chassis configuration to provide additional access ports as your office grows.

This example describes how to configure an EX4500 Virtual Chassis with a primary and backup in a single wiring closet:

### Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later for EX Series switches

**NOTE:** You must use Junos OS Release 11.4 or later if you are including three or more EX4500 switches in an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis.

- Two EX4500 switches with Virtual Chassis modules

Before you begin, be sure you have:

1. Rack-mounted the switches. See [Mounting an EX4500 Switch](#).
2. Cabled the switches but not the Virtual Chassis ports (VCPs).
3. Installed the same version of Junos OS Release 11.1 or later on both the switches

## Overview and Topology

### IN THIS SECTION

- [Topology | 107](#)

A Virtual Chassis configuration allows you to accommodate the networking needs of a growing office. The default configuration of a two-member Virtual Chassis includes a primary and a backup switch. In addition to providing more access ports than a single switch can provide, a Virtual Chassis configuration provides high availability through redundancy.

After you interconnect the switches using the dedicated VCPs and power on the switches, the VCPs are operational. The primary role priorities and member IDs are assigned by the software. The software elects a primary based on several criteria, including how long a member switch has belonged to the Virtual Chassis configuration. For additional details, see *Understanding How the Primary in a Virtual Chassis Is Elected*. Therefore, we recommend that you start by powering on only one member switch, the one that you want to function as the primary.

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a Virtual Chassis.

The Virtual Chassis configuration provides networking access for onsite workers who are sitting within the range of a single wiring closet. The workers all use personal computers and VoIP phones. As the office grows, you can add EX4200 switches to meet increased needs for access ports.

Topology

The topology for this example consists of two EX4500 switches.

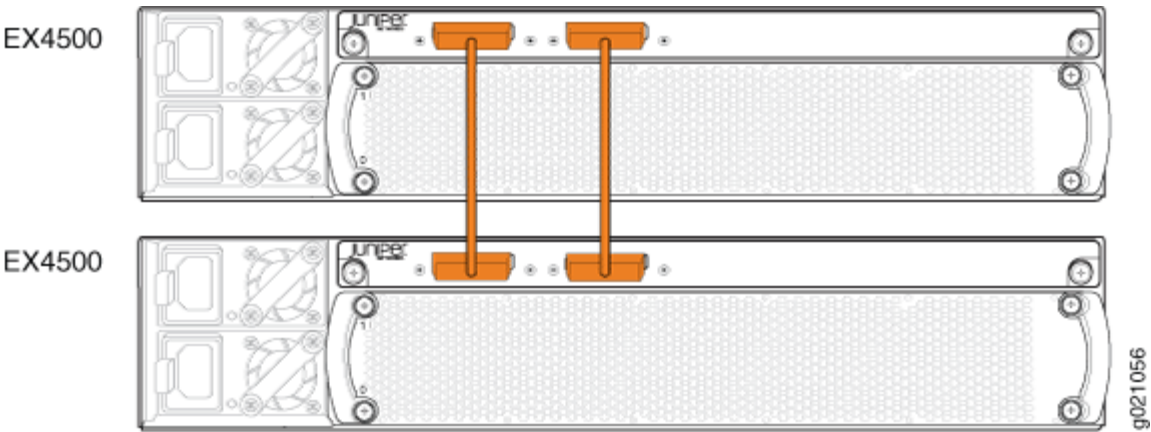
[Table 12 on page 107](#) shows the default configuration settings for the two-member Virtual Chassis.

**Table 12: Components of the Basic Virtual Chassis Access Switch Topology**

Member Switch	Hardware	Member ID	Role and Priority
SWA-0	EX4500 switch	0	Primary: primary-role priority 128
SWA-1	EX4500 switch	1	Backup: primary-role priority 128

[Figure 9 on page 107](#) shows that switches SWA-0 and SWA-1 are interconnected with their dedicated VCPs on the rear panel. The LCD on the front displays the member ID and role.

**Figure 9: Basic EX4500 Virtual Chassis with Primary and Backup**



Configuration

IN THIS SECTION

- Procedure | 108



Configure a Virtual Chassis with a default primary and backup in a single wiring closet:

## Procedure

### Step-by-Step Procedure

To configure a Virtual Chassis with primary and backup:

1. Power on SWA-0 (the member switch that you want to function as the primary).
2. Power on SWA-1 (the member switch that you want to function as the backup).
3. Set the PIC mode to Virtual Chassis mode on both switches:

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

4. Run the EZSetup program on SWA-0, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.
5. Cable the Virtual Chassis member switches together.
6. (Optional) Configure SWA-0 with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis configuration, if desired:

```
[edit]
user@SWA-0# set interfaces vme unit 0 family inet address /ip-address/mask/
```

7. (Optional, but recommended) Disable the split and merge feature:

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

## Verification

### IN THIS SECTION

- [Verifying That the Primary-role Priority Is Assigned Appropriately | 109](#)
- [Verifying That the VCPs Are Operational | 110](#)

To confirm that the Virtual Chassis configuration is operational, perform these tasks:

## Verifying That the Primary-role Priority Is Assigned Appropriately

### Purpose

Verify that the primary, which has been selected by default, is the member switch that you want to function in that role.

### Action

1. Check the front-panel LCD to confirm that the switch has powered on correctly and that a member ID has been assigned.
2. List the member switches of the Virtual Chassis configuration.

```
user@SWA-0> show virtual-chassis
```

```
Virtual Chassis ID: 0019.e250.47a0
```

				Mastership		Neighbor List	
Member ID	Status	Serial No	Model	priority	Role	ID	Interface
0 (FPC 0)	Prsnt	AK0207360276	ex4500-40f	128	Master*	1	vcp-0
						1	vcp-1
1 (FPC 1)	Prsnt	AK0207360281	ex4500-40f	128	Backup	0	vcp-0
						0	vcp-1

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

### Meaning

The `show virtual-chassis` command lists the member switches interconnected in a Virtual Chassis configuration with the member IDs that have been assigned by the primary, the primary-role priority values, and the roles. It also displays the neighbor members with which each member is interconnected. The output shows that SWA-0, member 0, has been assigned default primary-role priority 128. Because SWA-0 is the first member to be powered on, it has the most seniority and is therefore assigned the role of primary. SWA-1 is powered on after member 0, so it is assigned the role of backup. The member IDs are displayed on the front panel of the switches. Check and confirm whether the default assignment is satisfactory.

## Verifying That the VCPs Are Operational

### Purpose

Verify that the dedicated VCPs interconnecting the switches are operational.

### Action

Display the VCPs of all the members:

```
user@SWA-0> show virtual-chassis vc-port all-members
```

```
fpc0:
```

```
-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID       Interface
PIC / Port
vcp-0          Dedicated Up       32000      1    vcp-1
vcp-1          Dedicated Up       32000      1    vcp-0
```

```
fpc1:
```

```
-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID       Interface
PIC / Port
vcp-0          Dedicated Up       32000      1    vcp-1
vcp-1          Dedicated Up       32000      1    vcp-0
```

### Meaning

The `show virtual-chassis` command lists the interfaces that are enabled for the member switches of the Virtual Chassis configuration and shows the status of the interfaces. The output in this example shows that two of the VCPs are operational. A single cable has been used to interconnect vcp-0 of member ID 0 and vcp-0 of member ID 1. That interconnection is sufficient for the switch to be operational. However, we recommend that you connect the second set of VCPs for redundancy.

## Troubleshooting the Virtual Chassis

### IN THIS SECTION

- [Troubleshooting the Assignment of Roles | 111](#)
- [Troubleshooting the VCPs | 111](#)

To troubleshoot the configuration of a Virtual Chassis, perform these tasks:

### Troubleshooting the Assignment of Roles

#### Problem

The primary and backup roles are not assigned to the member switches that you want to function in these roles.

#### Solution

Modify the primary-role priority values.

To quickly modify the primary-role priority of SWA-1 (member ID 1), copy the following command and paste it into the switch terminal window:

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

### Troubleshooting the VCPs

#### Problem

The VCPs are down.

#### Solution

1. Check to make sure that you have cabled the appropriate ports.
2. Check to make sure that the cables are seated properly.

We recommend that you interconnect both of the VCPs on the member switches, for redundancy and high availability.

## RELATED DOCUMENTATION

[Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets | 159](#)

[Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis | 183](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

## Example: Expanding an EX3300 Virtual Chassis

### IN THIS SECTION

- [Requirements | 112](#)
- [Overview and Topology | 113](#)
- [Configuration | 114](#)
- [Verification | 116](#)
- [Troubleshooting | 118](#)

An EX3300 Virtual Chassis is a scalable switch composed of multiple interconnected EX3300 switches.

This example describes how to configure an expanding EX3300 Virtual Chassis within a single wiring closet. The process, however, would be identical if you were connecting EX3300 member switches in different wiring closets:

### Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.3 or later for EX Series switches
- Three EX3300 switches

Before you begin, be sure you have confirmed that the existing EX3300 Virtual Chassis configuration is operating correctly. See *Verifying That Virtual Chassis Ports Are Operational*.

## Overview and Topology

IN THIS SECTION

- [Topology | 113](#)

A Virtual Chassis configuration can be expanded without disrupting the site's network connectivity. This example describes adding a member switch to an existing Virtual Chassis configuration to provide additional access ports for connecting more PCs and voice over IP (VoIP) phones at this location. You can continue to expand the Virtual Chassis configuration with additional members in the same wiring closet, using the same procedure.

If you want to retain the roles of the existing primary and backup switches, explicitly configure the primary-role priority of these switches, specifying the highest possible value (255) for both the primary and the backup.

During expansion, the existing Virtual Chassis configuration can remain powered on and connected to the network. Before powering up the new switch, interconnect it to the other switches using the default VCP uplink ports on the front panel. Do not run the EZSetup program on the added member switch.

This example shows an existing Virtual Chassis configuration composed of two EX3300 switches. The Virtual Chassis configuration is being expanded to include an additional EX3300 switch.

### Topology

The topology for this example consists of three EX3300 switches.

[Table 13 on page 113](#) shows the configuration settings for the expanded Virtual Chassis.

**Table 13: Components of the Expanded Virtual Chassis Access Switch**

Member Switch	Hardware	Member ID	Role in Virtual Chassis
SWA-0	EX3300 switch	0	primary; primary-role priority 255
SWA-1	EX3300 switch	1	backup; primary-role priority 255

Table 13: Components of the Expanded Virtual Chassis Access Switch *(Continued)*

Member Switch	Hardware	Member ID	Role in Virtual Chassis
SWA-2	EX3300 switch	2	linecard; primary-role priority 128

## Configuration

### IN THIS SECTION

- [Procedure | 114](#)

To expand a Virtual Chassis configuration to include additional member switches within a single wiring closet, perform these tasks:

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis configuration.

## Procedure

### CLI Quick Configuration

To maintain the primary and backup roles of the existing members and ensure that the new member switch functions in a linecard role, copy the following commands and paste them into the terminal window:

```
[edit]
user@SWA-0# set virtual-chassis member 0 mastership-priority 255
user@SWA-1# set virtual-chassis member 1 mastership-priority 255
```

## Step-by-Step Procedure

To ensure that the existing member switches retain their current roles and to add another member switch in a linecard role:

1. Configure the primary-role priority of SWA-0 (member 0) to be the highest possible value, thereby ensuring that it functions as the primary of the expanded Virtual Chassis configuration.

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

2. Configure the primary-role priority of SWA-1 (member 1) to be the highest possible value. This setting is recommended for high availability and smooth transition of primary role in case the original primary becomes unavailable.

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

3. Interconnect the unpowered SWA-2 with SWA-0 using the default VCPs on the front panel. See [Virtual Chassis Cabling Configuration Examples for EX3300 Switches](#) for additional information.

4. Power on SWA-2.

You do not need to run EZSetup on SWA-2. The identification parameters that were set up for the primary apply implicitly to all members of the Virtual Chassis configuration. SWA-2 functions in a linecard role, because SWA-0 and SWA-1 have been configured to the highest primary-role priority values.

5. Confirm SWA-2 is now included in the Virtual Chassis configuration by checking the front-panel LCD for the member ID of this switch.

6. Cable the other VCP on SWA-2 to the Virtual Chassis.

**NOTE:** If you immediately cable both VCPs on SWA-2 into the existing Virtual Chassis, SWA-0 or SWA-1 might become nonoperational for several seconds. Network traffic to one of the switches is dropped during the downtime.

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





Meaning

The `show virtual-chassis` command lists the member switches of the Virtual Chassis configuration with the member IDs and primary-role priority values. It also displays the neighbor members with which each member is interconnected. This output shows that SWA-2 has been assigned member ID 2 and has the default primary-role priority value 128. Because the primary-role priority is lower than the primary-role priority of the other members, SWA-2 functions in the linecard role. You can continue to add more member switches, following the same procedure. It is possible to have multiple members in linecard roles with the same primary-role priority value.

Verifying That the VCPs Are Operational

Purpose

Verify that the VCPs interconnecting the member switches are operational.

Action

List the VCP interfaces on the Virtual Chassis configuration.

```
user@SWA-0>show virtual-chassis vc-port all-members

fpc0:
-----
Interface      Type      Status
or
PIC / Port
1/2            Configured  Up
1/3            Configured  Up

fpc1:
-----
Interface      Type      Status
or
PIC / Port
1/2            Configured  Up
1/3            Configured  Up

fpc2:
-----
Interface      Type      Status
```

or		
PIC / Port		
1/2	Configured	Up
1/3	Configured	Up

Meaning

The `show virtual-chassis vc-port all-members` command lists all the interfaces for the Virtual Chassis configuration. In this case, no VCPs have been configured. However, the VCPs are automatically configured and enabled when you interconnect member switches using the default VCP uplink ports. We recommend that you interconnect the member switches using both VCPs for redundancy. The VCP interfaces are identified simply as **vcp-0** and **vcp-1**. The **fpc** number is the same as the member ID.

Troubleshooting

IN THIS SECTION

- [Troubleshooting Primary-role Priority | 118](#)
- [Troubleshooting Nonoperational VCPs | 119](#)

To troubleshoot the configuration of an expanded Virtual Chassis, perform these tasks:

Troubleshooting Primary-role Priority

Problem

You want to designate a different member as the primary.

Solution

Change the primary-role priority value or values of the switches, designating the highest primary-role priority value for the switch that you want to be primary.

1. Lower the primary-role priority of the existing primary (member 0).

```
[edit virtual-chassis]
user@SWA-0# set member 0 mastership-priority 1
```

2. Set the primary-role priority of the member that you want to be the primary to the highest possible value (255):

```
[edit virtual-chassis]
user@SWA-2# set member 2 mastership-priority 255
```

## Troubleshooting Nonoperational VCPs

### Problem

The VCP interface shows a status of **down**.

### Solution

Check the cable to make sure that it is properly and securely connected to the VCPs.

## RELATED DOCUMENTATION

| [Configuring an EX3300 Virtual Chassis \(CLI Procedure\)](#) | 240

## Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet

### IN THIS SECTION

- [Requirements](#) | 120
- [Overview and Topology](#) | 120
- [Configuration](#) | 122

- Verification | 124
- Troubleshooting | 126

A Virtual Chassis is a scalable switch composed of multiple interconnected EX4200 and EX4500 switches.

This example describes how to configure an expanding EX4200 Virtual Chassis within a single wiring closet:

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.0 or later for EX Series switches
- One EX4200-48P switch
- One EX4200-24T switch
- One EX4200-24P switch
- One XFP uplink module

Before you begin, be sure you have confirmed that the existing EX4200 Virtual Chassis configuration is operating correctly. See *Verifying That Virtual Chassis Ports Are Operational*.

## Overview and Topology

### IN THIS SECTION

- Topology | 121

A Virtual Chassis configuration can be expanded without disrupting the site's network connectivity. This example describes adding a member switch to an existing Virtual Chassis configuration to provide additional access ports for connecting more PCs and Voice over IP (VoIP) phones at this location. You can continue to expand the Virtual Chassis configuration with additional members in the same wiring closet, using the same procedure. If you want to expand the Virtual Chassis configuration to include member switches in another wiring closet, see ["Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets"](#) on page 148.

If you want to retain the roles of the existing primary and backup switches, explicitly configure the primary-role priority of these switches, specifying the highest possible value (255) for both the primary and the backup.

During expansion, the existing Virtual Chassis configuration can remain powered on and connected to the network. Before powering up the new switch, interconnect it to the other switches using the dedicated VCPs on the rear panel. Do not run the EZSetup program on the added member switch.

This example shows an existing Virtual Chassis configuration composed of two EX4200 switches. The Virtual Chassis configuration is being expanded to include an EX4200-24P switch.

## Topology

The topology for this example consists of:

- One EX4200-48P switch (SWA-0) with 48 access ports, all of which support Power over Ethernet (PoE)
- One EX4200-24T switch (SWA-1) with 24 access ports, including eight ports that support PoE
- One EX4200-24P switch (SWA-2) with 24 access ports, all of which support PoE
- One uplink module with two 10-gigabit ports is installed in the EX4200-48P switch. These ports can be configured as trunk ports to connect to a distribution switch or customer edge (CE) router or as Virtual Chassis ports (VCPs) to interconnect with a member switch that is located too far for dedicated VCP cabling. (The uplink module ports on the SFP and SFP+ uplink modules and the SFP network interfaces on the EX4200-24F switches can also be used for these purposes.) For information on configuring the uplink ports as trunk ports to a distribution switch, see [Configuring Gigabit Ethernet Interfaces \(CLI Procedure\)](#). For information on configuring uplink ports as VCPs, see *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*.

[Table 14 on page 121](#) shows the configuration settings for the expanded Virtual Chassis.

**Table 14: Components of the Expanded Virtual Chassis Access Switch**

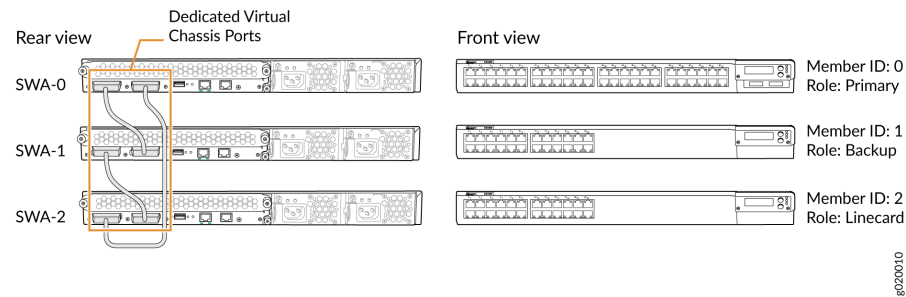
Member Switch	Hardware	Member ID	Role in Virtual Chassis
SWA-0	EX4200-48P switch	0	primary; primary-role priority 255
SWA-1	EX4200-24T switch	1	backup; primary-role priority 255

Table 14: Components of the Expanded Virtual Chassis Access Switch *(Continued)*

Member Switch	Hardware	Member ID	Role in Virtual Chassis
SWA-2	EX4200-24P switch	2	linecard; primary-role priority 128

Figure 10 on page 122 shows that the three member switches ( SWA-0, SWA-1 and SWA-2) are interconnected with their dedicated VCPs on the rear panel. The LCD on the front displays the member ID and role. SWA-0 also includes an uplink module. Its uplink ports can be used to connect to a distribution switch.

Figure 10: Expanded EX4200 Virtual Chassis in a Single Wiring Closet



## Configuration

### IN THIS SECTION

- [Procedure | 123](#)

To expand a Virtual Chassis configuration to include additional member switches within a single wiring closet, perform these tasks:

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis configuration.

## Procedure

### CLI Quick Configuration

To maintain the primary and backup roles of the existing members and ensure that the new member switch functions in a linecard role, copy the following commands and paste them into the terminal window:

```
[edit]
user@SWA-0# set virtual-chassis member 0 mastership-priority 255
user@SWA-1# set virtual-chassis member 1 mastership-priority 255
```

### Step-by-Step Procedure

To ensure that the existing member switches retain their current roles and to add another member switch in a linecard role:

1. Configure the primary-role priority of SWA-0 (member 0) to be the highest possible value, thereby ensuring that it functions as the primary of the expanded Virtual Chassis configuration.

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

2. Configure the primary-role priority of SWA-1 (member 1) to be the highest possible value. This setting is recommended for high availability and smooth transition of primary role in case the original primary becomes unavailable.

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

3. Interconnect the unpowered SWA-2 with SWA-0 using the dedicated VCPs on the rear panel. See *Virtual Chassis Cabling Configuration Examples for EX4200 Switches* for additional information.
4. Power on SWA-2.

You do not need to run EZSetup on SWA-2. The identification parameters that were set up for the primary apply implicitly to all members of the Virtual Chassis configuration. SWA-2 functions in a linecard role, because SWA-0 and SWA-1 have been configured to the highest primary-role priority values.



5. Confirm SWA-2 is now included in the Virtual Chassis configuration by checking the front-panel LCD for the member ID of this switch.
6. Cable the other VCP on SWA-2 to the Virtual Chassis.

**NOTE:** If you immediately cable both VCPs on SWA-2 into the existing Virtual Chassis, SWA-0 or SWA-1 might become nonoperational for several seconds. Network traffic to one of the switches is dropped during the downtime.

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

## Verification

### IN THIS SECTION

- [Verifying That the New Switch Has Been Added in a Linecard Role | 124](#)
- [Verifying That the VCPs Are Operational | 125](#)

To verify that the new switch has been added in the linecard role and that its VCPs are operational, perform these tasks:

### Verifying That the New Switch Has Been Added in a Linecard Role

#### Purpose

Verify that SWA-2 has been added in a linecard role to the Virtual Chassis configuration.

#### Action

Use the `show virtual-chassis` command to list the member switches with their member IDs, primary-role priority values, and assigned roles.

```
user@SWA-0> show virtual-chassis
```

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

```
Mastership
```

```
Neighbor List
```

Member ID	Status	Serial No	Model	Priority	Role	ID Interface
0 (FPC 0)	Prsnt	abc123	ex4200-48p	255	Master*	1 vcp-0 2 vcp-1
1 (FPC 1)	Prsnt	def456	ex4200-24t	255	Backup	2 vcp-0 0 vcp-1
2 (FPC 2)	Prsnt	abd231	ex4200-24p	128	Linecard	0 vcp-0 1 vcp-1

## Meaning

The `show virtual-chassis` command lists the member switches of the Virtual Chassis configuration with the member IDs and primary-role priority values. It also displays the neighbor members with which each member is interconnected. This output shows that SWA-2 has been assigned member ID 2 and has the default primary-role priority value 128. Because the primary-role priority is lower than the primary-role priority of the other members, SWA-2 functions in the linecard role. You can continue to add more member switches, following the same procedure. It is possible to have multiple members in linecard roles with the same primary-role priority value.

## Verifying That the VCPs Are Operational

### Purpose

Verify that the dedicated VCPs interconnecting the member switches are operational.

### Action

List the VCP interfaces on the Virtual Chassis configuration.

```
user@SWA-0>show virtual-chassis vc-port all-members
```

```
fpc0:
```

```
-----
```

Interface	Type	Status
or PIC / Port		
vcp-0	Dedicated	Up
vcp-1	Dedicated	Up

fpc1:  
-----

Interface	Type	Status
or PIC / Port		
vcp-0	Dedicated	Up
vcp-1	Dedicated	Up

fpc2:  
-----

Interface	Type	Status
or PIC / Port		
vcp-0	Dedicated	Up
vcp-1	Dedicated	Up

Meaning

The `show virtual-chassis vc-port all-members` command lists all the interfaces for the Virtual Chassis configuration. In this case, no VCP uplinks have been configured. However, the VCP interfaces are automatically configured and enabled when you interconnect member switches using the dedicated VCPs. We recommend that you interconnect the member switches using both VCPs for redundancy. The VCP interfaces are identified simply as **vcp-0** and **vcp-1**. The **fpc** number is the same as the member ID.

Troubleshooting

IN THIS SECTION

[Troubleshooting Primary-role Priority | 127](#)

[Troubleshooting Nonoperational VCPs | 127](#)

To troubleshoot the configuration of an expanded Virtual Chassis, perform these tasks:

## Troubleshooting Primary-role Priority

### Problem

You want to designate a different member as the primary.

### Solution

Change the primary-role priority value or values of the switches, designating the highest primary-role priority value for the switch that you want to be primary.

1. Lower the primary-role priority of the existing primary (member 0).

```
[edit virtual-chassis]
user@SWA-0# set member 0 mastership-priority 1
```

2. Set the primary-role priority of the member that you want to be the primary to the highest possible value (255):

```
[edit virtual-chassis]
user@SWA-2# set member 2 mastership-priority 255
```

## Troubleshooting Nonoperational VCPs

### Problem

The VCP interface shows a status of **down**.

### Solution

Check the cable to make sure that it is properly and securely connected to the VCPs.

## RELATED DOCUMENTATION

[Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration | 140](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

## Example: Adding EX4500 Switches to a Preprovisioned Virtual Chassis

### IN THIS SECTION

- [Requirements | 128](#)
- [Overview and Topology | 129](#)
- [Configuration | 130](#)
- [Verification | 132](#)

A Virtual Chassis is multiple switches operating as a single network entity. You might want to expand your existing Virtual Chassis by adding EX4500 switches to your Virtual Chassis configuration. You can include up to ten EX4500 switches in an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis.

This example describes how to add two EX4500 switches to an existing EX4200 Virtual Chassis that was preprovisioned. A preprovisioned configuration is a Virtual Chassis configuration in which the roles are statically assigned to each member switch:

### Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later for EX Series switches
- Two EX4200 switches interconnected into a Virtual Chassis
- Two standalone EX4500 switches with Virtual Chassis modules

Before you begin, be sure you have:

1. An operational EX4200 Virtual Chassis with two member EX4200 switches that was configured using a preprovisioned configuration.
2. Installed the same version of Junos OS Release 11.1 or later on all members of the Virtual Chassis and on the EX4500 switches.

**NOTE:** You must use Junos OS Release 11.4 or later if you are including three or more EX4500 switches in an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis.

## Overview and Topology

### IN THIS SECTION

- [Topology | 129](#)

You can create an EX4200 Virtual Chassis by cabling two operational EX4200 switches together using the dedicated Virtual Chassis ports (VCPs) on each EX4200 switch. When you preprovision a Virtual Chassis, you configure the roles for each member switch.

### Topology

[Table 15 on page 129](#) shows the configuration of the two-member EX4200 Virtual Chassis before the EX4500 member switches were added.

**Table 15: Components of the EX4200 Virtual Chassis Before the EX4500 Member Switches Are Added**

Member Switch	Hardware	Member ID	Role and Priority
SWA-0	EX4200 switch	0	Primary
SWA-1	EX4200 switch	1	Backup

[Table 16 on page 129](#) shows the configuration of the mixed EX4200 and EX4500 Virtual Chassis after the EX4500 member switches are added to the Virtual Chassis and the roles are assigned.

**Table 16: Final Mixed EX4200 and EX4500 Virtual Chassis Components**

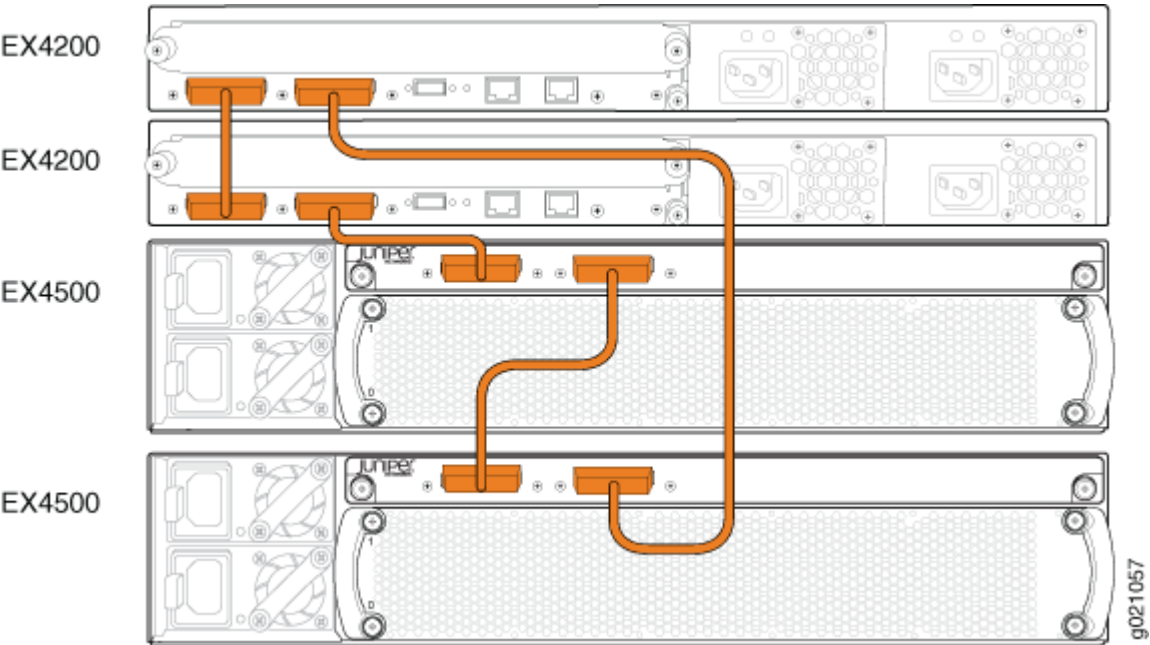
Member Switch	Hardware	Member ID	Role and Priority
SWA-0	EX4200 switch	0	Linecard

Table 16: Final Mixed EX4200 and EX4500 Virtual Chassis Components (Continued)

Member Switch	Hardware	Member ID	Role and Priority
SWA-1	EX4200 switch	1	Linecard
SWA-2	EX4500 switch	2	Primary
SWA-3	EX4500 switch	3	Backup

Figure 11 on page 130 shows the hardware topology of the mixed EX4200 and EX4500 Virtual Chassis.

Figure 11: Mixed EX4200 and EX4500 Virtual Chassis Topology (Preprovisioned Configuration)



Configuration

IN THIS SECTION

Procedure | 131

## Procedure

### Step-by-Step Procedure

To add two EX4500 switches to an existing EX4200 Virtual Chassis that was preprovisioned:

1. Log in to the EX4200 Virtual Chassis.
2. Set all member switches into mixed mode:

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

3. Reboot all member switches in the Virtual Chassis:

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

4. Power on SWA-2 and SWA-3, the EX4500 switches.
5. Set the PIC mode to the Virtual Chassis mode on both SWA-2 and SWA-3:

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

6. On each of the switches SWA-2 and SWA-3, configure the switch as a member of the mixed EX4200 and EX4500 Virtual Chassis:

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

7. Reboot the EX4500 switches.
8. Log in to the EX4200 Virtual Chassis and change the roles of the EX4200 member switches to **line-card**:

```
[edit virtual-chassis]
user@SWA-0# set member 0 role line-card
user@SWA-0# set member 1 role line-card
```



9. Add the EX4500 switches to the Virtual Chassis configuration:

```
[edit virtual-chassis]
user@SWA-0# set member 2 serial-number ghi789 role routing-engine
user@SWA-0# set member 3 serial-number jkl012 role routing-engine
```

10. Cable SWA-2 and SWA-3 into the Virtual Chassis using the dedicated VCPs.

**NOTE:** We recommend cabling one VCP into the Virtual Chassis, waiting for the new switch to be recognized by the Virtual Chassis, then cabling the other VCP on the new switch into the Virtual Chassis.

If you immediately cable both VCPs into the existing Virtual Chassis, one of the Virtual Chassis member switches might become nonoperational for several seconds. Network traffic to that member switch is dropped during the downtime.

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

## Verification

### IN THIS SECTION

- [Verifying Virtual Chassis Availability and Roles | 132](#)

To confirm that the Virtual Chassis configuration is operational, perform these tasks:

### Verifying Virtual Chassis Availability and Roles

#### Purpose

Verify that the Virtual Chassis is up and that the member switches are in the correct roles.

Action

List the member switches and roles.

```
user@SWA-2> show virtual-chassis

Virtual Chassis ID: 0019.e250.47a0

Member ID  Status  Serial No  Model  Mastership  Role  Neighbor List
         ID              priority    ID  Interface
0 (FPC 0)  Prsnt   AK02073602764 ex4200-48p    0  Linecard    1  vcp-0
                                         3  vcp-1
1 (FPC 1)  Prsnt   AK02073602814 ex4200-48p    0  Linecard    0  vcp-0
                                         2  vcp-1
2 (FPC 2)  Prsnt   AK02073602844 ex4500-40f   129 Master*    1  vcp-0
                                         3  vcp-1
3 (FPC 3)  Prsnt   AK02073602088 ex4500-40f   129 Backup    2  vcp-0
                                         0  vcp-1
```

Meaning

The `show virtual-chassis` command lists the member switches interconnected in a Virtual Chassis configuration with the member IDs that have been assigned by the primary, the primary-role priority values, and the roles. It also displays the neighbor members with which each member is interconnected. The output shows that the switches have the correct primary role priorities and roles.

RELATED DOCUMENTATION

- [Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets | 159](#)
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 271](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

## Example: Adding an EX4500 Switch to a Nonprovisioned Virtual Chassis

### IN THIS SECTION

- [Requirements | 134](#)
- [Overview and Topology | 135](#)
- [Configuration | 137](#)
- [Verification | 139](#)

A Virtual Chassis is multiple switches operating as a single network entity. You might want to expand an existing Virtual Chassis by adding EX4500 switches to it. You can include up to ten EX4500 switches in an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis.

This example describes how to add two EX4500 switches to an existing EX4200 Virtual Chassis that was nonprovisioned. A nonprovisioned configuration is a Virtual Chassis whose roles were assigned automatically rather than configured statically (preprovisioned).

### Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later for EX Series switches
- Two EX4200 switches interconnected into a nonprovisioned Virtual Chassis
- Two standalone EX4500 switches with Virtual Chassis modules

Before you begin, be sure you have:

1. An operational EX4200 Virtual Chassis with two member EX4200 switches that was configured using a nonprovisioned configuration.
2. Installed the same version of Junos OS Release 11.1 or later for all members of the Virtual Chassis and on the EX4500 switches.

**NOTE:** You must use Junos OS Release 11.4 or later if you are including three or more EX4500 switches in an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis.

## Overview and Topology

### IN THIS SECTION

- [Topology | 135](#)

You can create an EX4200 Virtual Chassis by cabling two operational EX4200 switches together using the dedicated Virtual Chassis ports (VCPs) on each EX4200 switch. When you cable a Virtual Chassis using the dedicated VCPs, the switches run a primary-role election algorithm that determines the Virtual Chassis roles for each member switch. The primary-role election algorithm first checks the primary-role priority ID. The primary-role priority ID is any number between 0 and 255. The switch with the higher primary-role priority ID is elected into the primary role and the other switch, unless it has been configured with a primary-role priority value of 0, is elected into the backup role.

The default primary-role priority ID for an EX4200 or EX4500 switch is 128. Both EX4200 switches are using this default primary-role priority ID in the operational EX4200 Virtual Chassis in this example.

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a Virtual Chassis.

### Topology

[Table 17 on page 135](#) shows the default configuration settings for the two-member Virtual Chassis before the EX4500 member switches are added to the Virtual Chassis.

**Table 17: Components of the EX4200 Virtual Chassis Before the EX4500 Member Switches Are Added**

Member Switch	Hardware	Member ID	Role and Priority
SWA-0	EX4200 switch	0	Primary: primary-role priority 128
SWA-1	EX4200 switch	1	Backup: primary-role priority 128

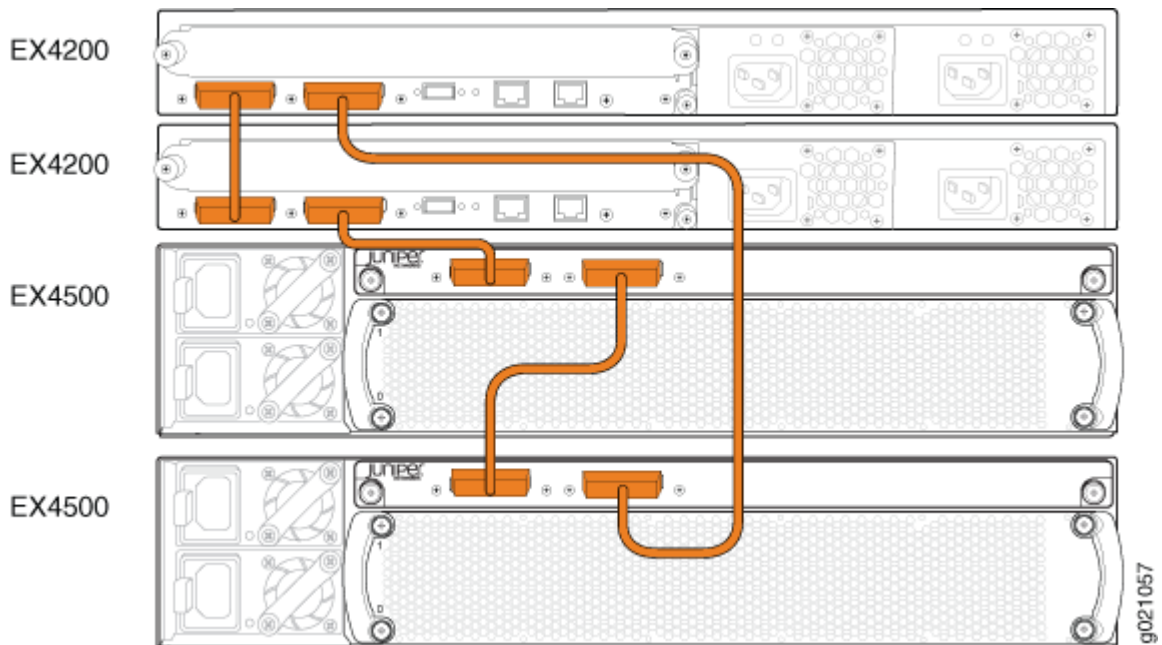
[Table 18 on page 136](#) shows the configuration of the mixed EX4200 and EX4500 Virtual Chassis after the EX4500 member switches are added to the Virtual Chassis and the EX4500 switches have been configured in the primary and backup roles.

Table 18: Final Mixed EX4200 and EX4500 Virtual Chassis Components

Member Switch	Hardware	Member ID	Role and Priority
SWA-0	EX4200 switch	0	Linecard: primary-role priority 0
SWA-1	EX4200 switch	1	Linecard: primary-role priority 0
SWA-2	EX4500 switch	2	Primary: primary-role priority 255
SWA-3	EX4500 switch	3	Backup: primary-role priority 255

Figure 12 on page 136 shows the hardware topology of the mixed EX4200 and EX4500 Virtual Chassis.

Figure 12: Mixed EX4200 and EX4500 Virtual Chassis Topology (Nonprovisioned Configuration)



## Configuration

### IN THIS SECTION

- [Procedure](#) | 137

### Procedure

#### Step-by-Step Procedure

To add two EX4500 switches to an existing EX4200 Virtual Chassis:

1. Log in to the EX4200 Virtual Chassis.
2. Set all member switches into mixed mode:

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

3. Reboot all member switches in the Virtual Chassis:

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

4. Power on SWA-2, the EX4500 switch that you want to function in the primary role.
5. Power on SWA-3, the EX4500 switch that you want to function in the backup role.
6. Set the PIC mode to the Virtual Chassis mode on both SWA-2 and SWA-3:

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

7. Configure SWA-2 and SWA-3 as a member of the mixed EX4200 and EX4500 Virtual Chassis:

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

8. Reboot the EX4500 switches.

9. Cable SWA-2 into the Virtual Chassis using the dedicated VCP on the back of the EX4200 member switch and the dedicated VCP on the Virtual Chassis module in the EX4500 switch.

**NOTE:** We recommend cabling one VCP into the Virtual Chassis, waiting for the new switch to be recognized by the Virtual Chassis, then cabling the other VCP on the new switch into the Virtual Chassis.

If you immediately cable both VCPs into the existing Virtual Chassis, one of the Virtual Chassis member switches might become nonoperational for several seconds. Network traffic to that member switch is dropped during the downtime.

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

10. Cable SWA-3 into the Virtual Chassis using the dedicated VCP on the back of the EX4200 member switch and the dedicated VCP on the Virtual Chassis module in the EX4500 switch.

**NOTE:** We recommend cabling one VCP into the Virtual Chassis, waiting for the new switch to be recognized by the Virtual Chassis, then cabling the other VCP on the new switch into the Virtual Chassis.

If you immediately cable both VCPs into the existing Virtual Chassis, one of the Virtual Chassis member switches might become nonoperational for several seconds. Network traffic to that member switch is dropped during the downtime.

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

11. Log in to the Virtual Chassis and set the primary-role priority for the EX4200 switches to **0** and the EX4500 switches to **255**:

```
[edit virtual-chassis]
user@SWA-0# set member 0 mastership-priority 0
user@SWA-0# set member 1 mastership-priority 0
user@SWA-0# set member 2 mastership-priority 255
user@SWA-0# set member 3 mastership-priority 255
```

**NOTE:** A switch with a primary-role priority value of 0 never assumes the primary or backup role within a Virtual Chassis. This configuration ensures that the EX4200 member switches remain in the linecard roles even during Virtual Chassis topology changes.

If you want the EX4200 switches to assume the primary or backup roles, assign the switches a primary-role priority between 1 and 255.

## Verification

### IN THIS SECTION

- [Verifying Virtual Chassis Availability and Roles | 139](#)

To confirm that the Virtual Chassis configuration is operational, perform these tasks:

### Verifying Virtual Chassis Availability and Roles

#### Purpose

Verify that the Virtual Chassis is up and that the member switches are in the correct roles.

#### Action

List the member switches and roles.

```
user@SWA-2> show virtual-chassis
```

Virtual Chassis ID: 0019.e250.47a0

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID Interface
0 (FPC 0)	Prsnt	AK02073602764	ex4200-48p	0	Linecard	1 vcp-0 3 vcp-1
1 (FPC 1)	Prsnt	AK02073602814	ex4200-48p	0	Linecard	0 vcp-0 2 vcp-1
2 (FPC 2)	Prsnt	AK02073602844	ex4500-40f	255	Master*	1 vcp-0 3 vcp-1
3 (FPC 3)	Prsnt	AK02073602088	ex4500-40f	255	Backup	2 vcp-0



0 vcp-1

Member ID for next new member: 4 (FPC 4)

## Meaning

The `show virtual-chassis` command lists the member switches interconnected in a Virtual Chassis configuration with the member IDs that have been assigned by the primary, the primary-role priority values, and the roles. It also displays the neighbor members with which each member is interconnected. The output shows that the switches have the correct primary role priorities and roles.

## RELATED DOCUMENTATION

[Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets | 159](#)

[Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 273](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

## Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration

### IN THIS SECTION

- [Requirements | 141](#)
- [Overview and Topology | 141](#)
- [Configuration | 142](#)
- [Verification | 143](#)
- [Troubleshooting | 146](#)

You can configure a multimember EX4200 Virtual Chassis access switch in a single wiring closet without setting any parameters—by simply cabling the switches together, using the dedicated Virtual Chassis ports (VCPs). You do not need to modify the default configuration to enable these ports. They are operational by default. The Virtual Chassis configuration automatically assigns the primary, backup, and

linecard roles, based on the sequence in which the switches are powered on and other factors in the primary-role election algorithm.

**TIP:** We recommend that you explicitly configure the primary-role priority of the switches to ensure that the switches continue to perform the desired roles when additional switches are added or other changes occur. However, it is possible to use the default configuration described in this example.

This example describes how to configure a multimember Virtual Chassis in a single wiring closet, using the default role assignments:

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.0 or later for EX Series switches
- Two EX4200-48P switches
- Four EX4200-24P switches

## Overview and Topology

### IN THIS SECTION

- [Topology | 141](#)

A Virtual Chassis configuration is easily expandable. This example shows a Virtual Chassis configuration composed of six EX4200 switches. It provides networking access for 180 onsite workers, who are sitting within range of a single wiring closet. The six combined switches are identified by a single host name and managed through a global management IP address.

To set up a multimember Virtual Chassis configuration within a single wiring closet, you need to run the EZSetup program only once. Connect to the primary and run EZSetup to specify its identification, time zone, and network properties. When additional switches are connected through the Virtual Chassis ports (VCPs), they automatically receive the same properties that were specified for the primary.

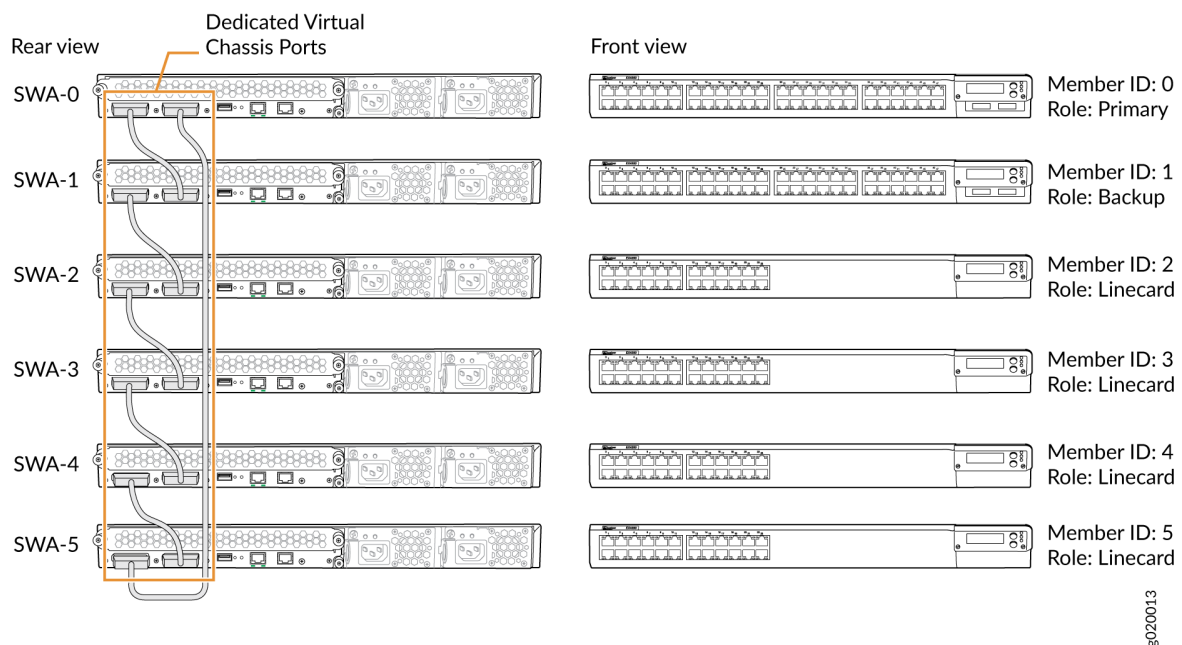
### Topology

The topology for this example (see [Figure 13 on page 142](#)) consists of six switches:

- Two EX4200-48P switches (SWA-0 and SWA-1) with 48 access ports, all of which support Power over Ethernet (PoE)
- Four EX4200-24P switches (SWA-2, SWA-3, SWA-4, and SWA-5) with 24 access ports, all of which support PoE

Figure 13 on page 142 shows that all the member switches are interconnected with the dedicated VCPs on the rear panel. The LCD on the front displays the member ID and role.

**Figure 13: Default Configuration of a Multimember EX4200 Virtual Chassis in a Single Wiring Closet**



## Configuration

### IN THIS SECTION

- Procedure | 143

To configure a multimember Virtual Chassis access switch in a single wiring closet using the factory defaults, perform this task:

## Procedure

### Step-by-Step Procedure

To configure a multimember Virtual Chassis with default role assignments:

1. Make sure the dedicated VCPs on the rear panel are properly cabled. See *Virtual Chassis Cabling Configuration Examples for EX4200 Switches* for additional information.
2. Power on the switch that you want to function as the primary (SWA-0). This example uses one of the larger switches (EX4200-48P) as the primary.
3. Check the front panel LCD to confirm that the switch has powered on correctly and that a member ID has been assigned.
4. Run the EZSetup program on SWA-0, the primary, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.
5. Configure SWA-0 with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis configuration, if desired.

[edit]

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

6. After a lapse of at least one minute, power on SWA-1. This example uses the second EX4200-48P switch as the backup.
7. Check the front panel LCD to confirm that the switch has powered on correctly and that a member ID has been assigned.
8. Power on SWA-2, and check the front panels to make sure that the switch is operating correctly.
9. Continue to power on the member switches one by one, checking the front panels as you proceed.

## Verification

### IN THIS SECTION

- [Verifying the Member IDs and Roles of the Member Switches | 144](#)
- [Verifying That the VCPs Are Operational | 145](#)

To confirm that the configuration is working properly, perform these tasks:

## Verifying the Member IDs and Roles of the Member Switches

### Purpose

Verify that all the interconnected member switches are included within the Virtual Chassis configuration and that their roles are assigned appropriately.

### Action

Display the members of the Virtual Chassis configuration:

```
user@SWA-0> show virtual-chassis
```

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

Member ID	Status	Serial No	Model	Mastership		Neighbor List
				Priority	Role	ID Interface
0 (FPC 0)	Prsnt	abc123	ex4200-48p	128	Master*	1 vcp-0 5 vcp-1
1 (FPC 1)	Prsnt	def123	ex4200-48p	128	Backup	2 vcp-0 0 vcp-1
2 (FPC 2)	Prsnt	abd231	ex4200-24p	128	Linecard	3 vcp-0 1 vcp-1
3 (FPC 3)	Prsnt	cab123	ex4200-24p	128	Linecard	4 vcp-0 2 vcp-1
4 (FPC 4)	Prsnt	fed456	ex4200-24p	128	Linecard	5 vcp-0 3 vcp-1
5 (FPC 5)	Prsnt	jk1231	ex4200-24p	128	Linecard	0 vcp-0 4 vcp-1

Meaning

The `show virtual-chassis` command lists the member switches of the Virtual Chassis configuration with the member IDs and primary-role priority values. It also displays the neighbor members with which each member is interconnected. The **fpc** number is the same as the member ID.

Verifying That the VCPs Are Operational

Purpose

Verify that the dedicated VCPs interconnecting the member switches are operational.

Action

Display the VCP interfaces.

```
user@SWA-0> show virtual-chassis vc-port all-members

fpc0:
-----
Interface      Type      Status
or
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up

fpc1:
-----
Interface      Type      Status
or
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up

fpc2:
-----
Interface      Type      Status
or
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up
```

fpc3:		
-----		
Interface	Type	Status
or		
PIC / Port		
vcp-0	Dedicated	Up
vcp-1	Dedicated	Up
fpc4:		
-----		
Interface	Type	Status
or		
PIC / Port		
vcp-0	Dedicated	Up
vcp-1	Dedicated	Up
fpc5:		
-----		
Interface	Type	Status
or		
PIC / Port		
vcp-0	Dedicated	Up
vcp-1	Dedicated	Up

### Meaning

The `show virtual-chassis vc-port all-members` command lists the VCP interfaces that are enabled for the member switches of the Virtual Chassis configuration and shows the status of the interfaces. In this case, no VCP uplinks have been configured. However, the VCP interfaces are automatically configured and enabled when you interconnect member switches using the dedicated VCPs. The dedicated VCP interfaces are identified simply as vcp-0 and vcp-1. They do not use the standard interface address (in which the member ID is represented by the first digit). The output in this example shows that all interfaces are operational. The **fpc** number is the same as the member ID.

### Troubleshooting

IN THIS SECTION

●

Troubleshooting Primary-role Priority | 147

●

Troubleshooting Nonoperational VCPs | 147

To troubleshoot the configuration of a multimember Virtual Chassis in a single wiring closet, perform these tasks:

## Troubleshooting Primary-role Priority

### Problem

You want to explicitly designate one member as the primary and another as backup.

### Solution

Change the primary-role priority value of the member that you want to function as primary, assigning the highest primary-role priority value to that member.

**NOTE:** These configuration changes are made through the current primary, SWA-0.

1. Configure primary-role priority of member 0 to be the highest possible value.

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

2. Set the primary-role priority of another member that you want to function as the backup member as the same value:

```
[edit virtual-chassis]
user@SWA-0# set member 2 mastership-priority 255
```

## Troubleshooting Nonoperational VCPs

### Problem

The VCP interface shows a status of **down**.

### Solution

Check the cable to make sure that it is properly and securely connected to the VCPs.



## RELATED DOCUMENTATION

*Understanding How the Primary in a Virtual Chassis Is Elected*

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

[Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets | 148](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

[Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)

## Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets

### IN THIS SECTION

- [Requirements | 149](#)
- [Overview and Topology | 149](#)
- [Configuration | 152](#)
- [Verification | 155](#)
- [Troubleshooting | 158](#)

An EX4200 Virtual Chassis can be composed of multiple EX4200 switches in different locations. You can install member switches in different wiring closets, interconnecting the member switches by cabling and configuring uplink module ports or SFP network ports on EX4200-24F switches as Virtual Chassis ports (VCPs).

This example shows how to use uplink VCPs to connect Virtual Chassis members that are located too far apart to be connected using the dedicated VCPs. Uplink VCPs can also be used to connect Virtual Chassis members to form link aggregation groups (LAGs). For the latter usage, see "[Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports](#)" on page 217.

**NOTE:** You can also configure the SFP network ports on EX4200-24F switches as VCPs to connect Virtual Chassis member switches across wiring closets and to form LAGs.

This example describes how to configure a Virtual Chassis access switch interconnected across wiring closets:

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.0 or later for EX Series switches
- Four EX4200 switches
- Four XFP uplink modules

Before you interconnect the members of the Virtual Chassis configuration across wiring closets, be sure you have:

1. Installed an uplink module in each member switch. See *Installing an Uplink Module in an EX4200 Switch*.
2. Powered on and connected the switch in the primary role, SWA-0, and run the EZSetup program (see [Table 19 on page 151](#) for switch names used in this example). See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.
3. Configured SWA-0 with the virtual management Ethernet (VME) interface for remote, out-of-band management of the Virtual Chassis configuration, if desired. See "[Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\)](#)" on page 295.
4. Interconnected SWA-0 and SWA-1 using the dedicated VCPs on the rear panel. SWA-1 must not be powered on at this time.
5. Interconnected SWA-2 and SWA-3 using the dedicated VCPs on the rear panel. SWA-2 and SWA-3 must not be powered on at this time.

## Overview and Topology

### IN THIS SECTION

- [Topology | 150](#)

In this example, four EX4200 switches will be interconnected in a Virtual Chassis configuration. Two of these switches (SWA-0 and SWA-1) are located in wiring closet A, and the two other switches (SWA-2 and SWA-3) are located in wiring closet B.

For ease of monitoring and manageability, we want to interconnect all four switches as members of a Virtual Chassis configuration. Prior to configuring the Virtual Chassis, we installed uplink modules in each of the member switches. In this example, uplink modules are installed in all four members so that there are redundant VCP connections across the wiring closets. If you want to expand this configuration to include more members within these wiring closets, you do not need to add any more uplink modules. Simply use the dedicated VCPs on the rear panel. The redundancy of uplink VCPs provided in this example is sufficient.

## Topology

We have interconnected the switches in wiring closet A and also interconnected the ones in wiring closet B using the dedicated VCPs. The interfaces for the dedicated VCPs are operational by default. They do not need to be configured.

However, the Virtual Chassis cables that interconnect the dedicated VCPs of member switches within a single wiring closet are not long enough to connect member switches across wiring closets. Instead, we will use the fiber-optic cable connections in the uplink modules to interconnect the member switches in wiring closet A to the member switches in wiring closet B. You only need to interconnect one member switch in wiring closet A to one in wiring closet B to form the Virtual Chassis configuration. However, for redundancy, this example connects uplink module ports from the two member switches in wiring closet A to the two member switches in wiring closet B.

We will specify the highest primary-role priority value (255) for SWA-0 to make it the primary before we power on SWA-1. Because SWA-0 and SWA-1 are interconnected with the dedicated VCPs, the primary detects that SWA-1 is a member of its Virtual Chassis configuration and assigns it a member ID.

We configure SWA-2 in wiring closet B without running EZSetup by directly connecting to the console port. If you wish, you can run EZSetup and specify identification parameters. Later, when you interconnect SWA-2 with SWA-0, the primary of the Virtual Chassis configuration, the primary overwrites any conflicting parameters.

We will use SWA-2 as the backup of the Virtual Chassis configuration. If a problem occurs in wiring closet A, SWA-2 would take control of the Virtual Chassis configuration and maintain the network connections. We will configure the same primary-role priority value for SWA-2 (255) that we configured for the primary. Because we power on SWA-0 before we power on SWA-2, SWA-0 has additional prioritization properties that allow it to retain the primary role of the Virtual Chassis configuration. We recommend setting identical primary-role priority values for the primary and backup members for high availability and smooth transition of primary role in case the original primary becomes unavailable. (Setting identical primary-role priority values for the primary and backup members prevents the previous primary from pre-empting the primary role from the new primary when the previous primary comes back online.)

After we have configured SWA-2 and set one of its uplink module ports as an uplink VCP, we will interconnect its uplink VCP with an uplink VCP on SWA-0.

Finally, we will power on SWA-3. Because SWA-3 is interconnected with SWA-2 using the dedicated VCPs on the rear panel, the primary will detect that SWA-3 is part of the expanded Virtual Chassis configuration and assign it member ID **3**. For redundancy, we will configure an uplink VCP on SWA-3 through the primary and interconnect that uplink VCP with an uplink VCP on SWA-1.

[Table 19 on page 151](#) shows the Virtual Chassis configuration settings for a Virtual Chassis composed of member switches in different wiring closets.

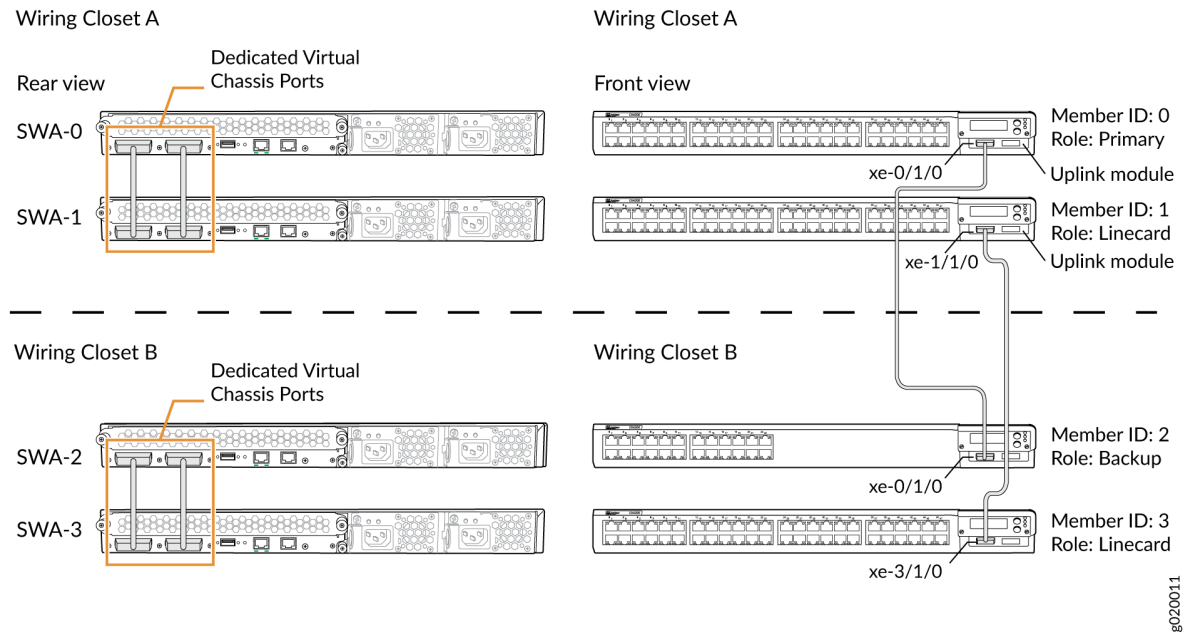
**Table 19: Components of a Virtual Chassis Interconnected Across Multiple Wiring Closets**

Switch	Member ID	Role and Priority	Location
SWA-0	0	primary; primary-role priority 255	Wiring closet A
SWA-1	1	linecard; primary-role priority 128	Wiring closet A
SWA-2	2	backup; primary-role priority 255	Wiring closet B
SWA-3	3	linecard; primary-role priority 128	Wiring closet B

[Figure 14 on page 152](#) shows the different types of interconnections used for this Virtual Chassis configuration. The rear view shows the member switches within each wiring closet interconnected to

each other using the dedicated VCPs. The front view shows the uplink VCPs interconnected across the wiring closets.

**Figure 14: EX4200 Virtual Chassis Interconnected Across Wiring Closets**



## Configuration

### IN THIS SECTION

- [Procedure | 152](#)

To configure the Virtual Chassis across multiple wiring closets, perform this task:

### Procedure

#### Step-by-Step Procedure

To configure a Virtual Chassis across multiple wiring closets:

1. Configure the primary-role priority of SWA-0 (member 0) to be the highest possible value (255), thereby ensuring that it functions as the primary of the expanded Virtual Chassis configuration:

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

2. Prepare the members in wiring closet A for interconnecting with the member switches in wiring closet B by setting uplink VCPs for member 0 and member 1:

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

**NOTE:**

- For redundancy, this example configures an uplink VCP in both SWA-0 and SWA-1.
- This example omits the specification of the **member** *member-id* option in configuring an uplink VCP for SWA-0 (and, later, for SWA-2). The command applies by default to the switch where it is executed.

3. Prepare SWA-2 in wiring closet B for interconnecting with the Virtual Chassis configuration by configuring its primary-role priority to be the highest possible value (255). Its member ID is currently 0, because it is not yet interconnected with the other members of the Virtual Chassis configuration. It is operating as a standalone switch. Its member ID will change when it is interconnected.

```
[edit virtual-chassis]
user@SWA-2# set member 0 mastership-priority 255
```

**NOTE:** SWA-2 is configured with the same primary-role priority value that we configured for SWA-0. However, the longer uptime of SWA-0 ensures that, once the interconnection is made, SWA-0 functions as the primary and SWA-2 functions as the backup.

4. Specify one uplink module port in SWA-2 as an uplink VCP. Its member ID is 0, because it is not yet interconnected with the other members of the Virtual Chassis configuration.

**NOTE:** The setting of the uplink VCP remains intact when SWA-2 reboots and joins the Virtual Chassis configuration as member 2.

```
user@SWA-2> request virtual-chassis vc-port set pic-slot 1 port 0
```

5. Physically interconnect SWA-0 and SWA-2 across wiring closets using their uplink VCPs. Although SWA-0 and SWA-2 have the same primary-role priority value (255), SWA-0 was powered on first and thus has longer uptime. This results in SWA-0 retaining primary role while SWA-2 reboots and joins the now expanded Virtual Chassis configuration as the backup, with member ID 2.
6. Power on SWA-3. It joins the expanded Virtual Chassis configuration as member 3.

**NOTE:** Member ID 3 is assigned to SWA-3 because SWA-3 was powered on after members 0, 1, and 2.

7. Because SWA-3 is now interconnected as a member of the Virtual Chassis configuration, you can specify a redundant uplink VCP on SWA-3 through the primary of the Virtual Chassis configuration:

```
user@SWA-0> request virtual-chassis vc-port set pic-slot 1 port 0 member
3
```

8. Physically interconnect SWA-3 and SWA-1 across wiring closets using their uplink VCPs. Both SWA-1 and SWA-3 have the default primary-role priority value (128) and function in a linecard role.

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis.

## Results

Display the results of the configuration on SWA-0:

```
[edit]
user@SWA-0# show virtual-chassis
```

```
member 0 {  
    mastership-priority 255;  
}  
member 1 {  
    mastership-priority 128;  
}  
member 2 {  
    mastership-priority 255;  
}  
member 3 {  
    mastership-priority 128;  
}  
}
```

## Verification

### IN THIS SECTION

- [Verifying the Member IDs and Roles of the Member Switches | 155](#)
- [Verifying that the Dedicated VCPs and Uplink VCPs Are Operational | 156](#)

To confirm that the configuration is working properly, perform these tasks:

### Verifying the Member IDs and Roles of the Member Switches

#### Purpose

Verify that all the interconnected member switches are included within the Virtual Chassis configuration and that their roles are assigned appropriately.

#### Action

Display the members of the Virtual Chassis configuration:

```
user@SWA-0> show virtual-chassis
```

```
Virtual Chassis ID: 000.e255.00e0
```



Member ID	Status	Serial No	Model	Mastership		Role	Neighbor List	
				Priority			ID	Interface
0 (FPC 0)	Prsnt	abc123	ex4200-48p	255		Master*	1 vcp-0	
							1 vcp-1	
							2 vcp-255/1/0	
1 (FPC 1)	Prsnt	def456	ex4200-24t	128		Linecard	0 vcp-0	
							0 vcp-1	
							3 vcp-255/1/0	
2 (FPC 2)	Prsnt	ghi789	ex4200-48p	255		Backup	3 vcp-0	
							3 vcp-1	
							0 vcp-255/1/0	
3 (FPC 3)	Prsnt	jkl012	ex4200-24t	128		Linecard	2 vcp-0	
							2 vcp-1	
							1 vcp-255/1/0	

## Meaning

The `show virtual-chassis` command lists the member switches interconnected as a Virtual Chassis configuration with the member IDs that have been assigned by the primary, the primary-role priority values, and the roles. It also displays the neighbor members with which each member is interconnected.

## Verifying that the Dedicated VCPs and Uplink VCPs Are Operational

### Purpose

Verify that the dedicated VCPs interconnecting member switches in wiring closet A and the uplink VCPs interconnecting the member switches between wiring closets are operational.

### Action

Display the VCP interfaces:

```
user@SWA-0> show virtual-chassis vc-port all-members
```

```
fpc0:
```

```

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

```

fpc1:

```

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

```

fpc2:

```

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

```

fpc3:

```

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

```

## Meaning

The dedicated VCPs are displayed as **vcp-0** and **vcp-1**. The interface on the switch that has been set as an uplink VCP is displayed as **1/0**. The member interface names of uplink VCPs are of the form **vcp-255/pic/port**—for example, **vcp-255/1/0**. In that name, **vcp-255** indicates that the interface is an

uplink VCP, **1** is the uplink PIC number, and **0** is the uplink port number. The **fpc** number is the same as the member ID. The **Trunk ID** is a positive number ID assigned to the LAG formed by the Virtual Chassis. If no LAG is formed, the value is **-1**.

## Troubleshooting

### IN THIS SECTION

- [Troubleshooting Nonoperational VCPs | 158](#)

To troubleshoot a Virtual Chassis configuration that is interconnected across wiring closets, perform these tasks:

### Troubleshooting Nonoperational VCPs

#### Problem

An uplink VCP shows a status of **down**.

#### Solution

- Check the cable to make sure that it is properly and securely connected to the ports.
- If the VCP is an uplink module port, make sure that it has been explicitly set as an uplink VCP.
- If the VCP is an uplink module port, make sure that you have specified the options (*pic-slot*, *port*, and *member*) correctly.

## RELATED DOCUMENTATION

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

[Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet | 119](#)

[Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration | 140](#)

*Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*

## Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets

### IN THIS SECTION

- Requirements | 159
- Overview and Topology | 160
- Configuration | 162
- Verification | 166
- Troubleshooting | 167

An EX4500 switch can be a member of an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis. An EX4500 Virtual Chassis can be composed of two to ten EX4500 switches in different wiring closets or locations. A mixed EX4200 and EX4500 Virtual Chassis can be composed of EX4200 and EX4500 switches in different locations or wiring closets provided that at least one EX4200 switch is connected to one EX4500 switch using the dedicated Virtual Chassis port (VCP) connections available on both switches.

You connect EX4500 member switches in a Virtual Chassis that are in different wiring closets by cabling them together using a 10-Gigabit Ethernet SFP+ connection. You then must configure the 10-Gigabit Ethernet SFP+ connection as a Virtual Chassis port (VCP).

This example shows how to use the 10-Gigabit Ethernet SFP+ ports on EX4500 switches to connect two EX4500 member switches that are located too far apart to be connected using the dedicated VCPs in a mixed EX4200 and EX4500 Virtual Chassis. The procedure to connect two EX4500 switches in an EX4500 Virtual Chassis is identical to the procedure shown in this example.

**NOTE:** Any 10-Gigabit Ethernet SFP+ connection on an EX4500 switch can be configured as a VCP. An EX4500 switch has network and uplink ports that support 10-Gigabit Ethernet SFP+ transceivers.

This example describes how to connect two EX4500 member switches across wiring closets:

### Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later for EX Series switches

**NOTE:** You must use Junos OS Release 11.4 or later if you are including three or more EX4500 switches in an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis.

- Two EX4500 member switches
- Four EX4200 member switches

Before you interconnect the members of the Virtual Chassis configuration across wiring closets, be sure you have:

1. Preprovisioned the Virtual Chassis. See ["Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)" on page 255](#) for details.
2. (Optional) Configured SWA-0 with the virtual management Ethernet (VME) interface if you want remote, out-of-band management of the Virtual Chassis configuration. See ["Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\)" on page 295](#).
3. Interconnected SWA-0, SWA-1, and SWA-2 using the dedicated VCPs.
4. Interconnected SWA-3, SWA-4, and SWA-5 using the dedicated VCPs.

## Overview and Topology

### IN THIS SECTION

- [Topology | 161](#)

In this example, two EX4500 switches and four EX4200 switches are interconnected in a mixed EX4200 and EX4500 Virtual Chassis configuration. One EX4500 switch (SWA-0) and two EX4200 switches (SWA-1 and SWA-2) are located in wiring closet A, and the other EX4500 switch (SWA-3) and the other two EX4200 switches (SWA-4 and SWA-5) are located in wiring closet B.

For ease of monitoring and manageability, we want to interconnect all six switches as members of a Virtual Chassis configuration.

## Topology

We have interconnected the switches in wiring closet A and also interconnected the ones in wiring closet B using the dedicated VCPs. The interfaces for the dedicated VCPs are operational by default. They do not need to be configured.

However, the Virtual Chassis cables that interconnect the dedicated VCPs of member switches within a single wiring closet are not long enough to connect member switches across wiring closets. Instead, we will use a 10-Gigabit Ethernet SFP+ connection to interconnect the member switches in wiring closet A to the member switches in wiring closet B. You only need to interconnect one member switch in wiring closet A to one in wiring closet B to form the Virtual Chassis configuration. In this example, this connection will be made by connecting the EX4500 switches in each wiring closet together by configuring a 10-Gigabit Ethernet SFP+ connection as a VCP.

We will preprovision the entire Virtual Chassis to set the roles for all member switches.

We will first power on SWA-0 and preprovision the Virtual Chassis. We will then cable the Virtual Chassis before powering on the other member switches.

[Table 20 on page 161](#) shows the Virtual Chassis configuration settings for a Virtual Chassis composed of member switches in different wiring closets.

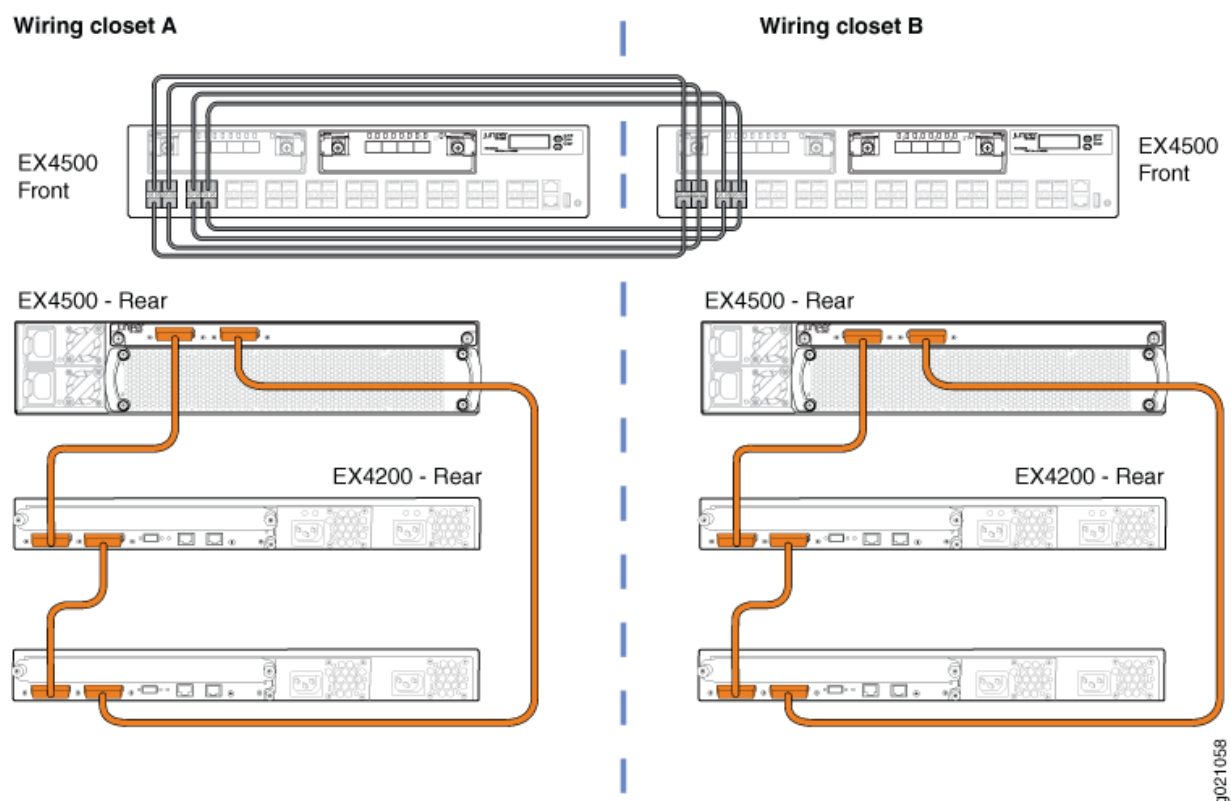
**Table 20: Components of a Virtual Chassis Interconnected Across Wiring Closets**

Switch	Model	Member ID	Role	Location
SWA-0	EX4500 switch	0	Primary	Wiring closet A
SWA-1	EX4200 switch	1	Linecard	Wiring closet A
SWA-2	EX4200 switch	2	Linecard	Wiring closet A
SWA-3	EX4500 switch	3	Backup	Wiring closet B
SWA-4	EX4200 switch	4	Linecard	Wiring closet B
SWA-5	EX4200 switch	5	Linecard	Wiring closet B

[Figure 15 on page 162](#) shows the different types of interconnections used for this Virtual Chassis configuration. The rear view shows the member switches within each wiring closet interconnected to

each other using the dedicated VCPs. The front view shows the uplink VCPs interconnected across the wiring closets.

**Figure 15: Mixed EX4200 and EX4500 Virtual Chassis Interconnected Across Wiring Closets**



## Configuration

### IN THIS SECTION

- [Procedure | 163](#)

To configure the Virtual Chassis across wiring closets, perform this task:

## Procedure

### Step-by-Step Procedure

To configure a Virtual Chassis across wiring closets:

1. Power on SWA-0 (the EX4500 switch acting as member 0).
2. Power on SWA-3 (the EX4500 switch acting as member 3).
3. Set the PIC mode to the Virtual Chassis mode on both SWA-0 and SWA-3:

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

4. Power on the remaining switches.
5. Configure all switches individually as members of the mixed EX4200 and EX4500 Virtual Chassis:

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

6. Reboot all switches:

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

7. Log back into SWA-0 after the reboot has completed.
8. Run EZSetup on SWA-0 to set the parameters for the entire Virtual Chassis. See *Connecting and Configuring an EX Series Switch (CLI Procedure)*.
9. Preprovision the Virtual Chassis from SWA-0. Specify all members for the Virtual Chassis configuration, listing each switch's serial number with the desired member ID and the desired role.

```
[edit virtual-chassis]
user@SWA-0# set preprovisioned
user@SWA-0# set member 0 serial-number abc123 role routing-engine
user@SWA-0# set member 1 serial-number def456 role line-card
user@SWA-0# set member 2 serial-number ghi789 role line-card
user@SWA-0# set member 3 serial-number jkl012 role routing-engine
user@SWA-0# set member 4 serial-number mno345 role line-card
user@SWA-0# set member 5 serial-number pqr678 role line-card
```



10. Commit the configuration:

```
user@SWA-0> commit synchronize
```

11. Prepare the members in wiring closet A for interconnecting with the member switches in wiring closet B by setting the 10-Gigabit Ethernet SFP+ interfaces on SWA-0 as VCPs:

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

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

user@SWA-0> request virtual-chassis vc-port set pic-slot 0 port 2

user@SWA-0> request virtual-chassis vc-port set pic-slot 0 port 3

user@SWA-0> request virtual-chassis vc-port set pic-slot 0 port 4

user@SWA-0> request virtual-chassis vc-port set pic-slot 0 port 5

user@SWA-0> request virtual-chassis vc-port set pic-slot 0 port 6

user@SWA-0> request virtual-chassis vc-port set pic-slot 0 port 7
```

12. Prepare the members in wiring closet B for interconnecting with the member switches in wiring closet A by setting the 10-Gigabit Ethernet SFP+ interfaces on SWA-3 as VCPs:

```
user@SWA-3> request virtual-chassis vc-port set pic-slot 0 port 0

user@SWA-3> request virtual-chassis vc-port set pic-slot 0 port 1

user@SWA-3> request virtual-chassis vc-port set pic-slot 0 port 2

user@SWA-3> request virtual-chassis vc-port set pic-slot 0 port 3

user@SWA-3> request virtual-chassis vc-port set pic-slot 0 port 4

user@SWA-3> request virtual-chassis vc-port set pic-slot 0 port 5

user@SWA-3> request virtual-chassis vc-port set pic-slot 0 port 6
```

```
user@SWA-3> request virtual-chassis vc-port set pic-slot 0 port 7
```

13. Physically interconnect SWA-0 with SWA-1, then interconnect all switches in wiring closet A.
14. Physically interconnect SWA-0 with SWA-3 across wiring closets using the 10-Gigabit Ethernet SFP + connections.
15. Physically interconnect all switches in wiring closet B.

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a Virtual Chassis.

## Results

Display the results of the configuration on SWA-0:

```
[edit virtual-chassis]
user@SWA-0# show
  member 0 {
    role routing-engine;
    serial-number abc123;
  }
  member 1 {
    role line-card;
    serial-number def456;
  }
  member 2 {
    role line-card;
    serial-number ghi789;
  }
  member 3 {
    role routing-engine;
    serial-number jkl012;
  }
  member 4 {
    role line-card;
    serial-number mno345;
  }
  member 5 {
```

```
        role line-card;
        serial-number pqr678;
    }
}
```

Verification

IN THIS SECTION

[Verifying the Member IDs and Roles of the Member Switches | 166](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying the Member IDs and Roles of the Member Switches

Purpose

Verify that all the interconnected member switches are included within the Virtual Chassis configuration and that their roles are assigned appropriately.

Action

Display the members of the Virtual Chassis configuration:

```
user@SWA-0> show virtual-chassis

Virtual Chassis ID: 000.e255.00e0
Virtual Chassis Mode: Mixed

      Mastership
Member ID  Status  Serial No  Model    Priority  Role    Neighbor List
      ID      Interface
0 (FPC 0)  Prsnt   abc123    ex4500-40f  255    Master*  1 vcp-0
                                     2 vcp-1
                                     3 vcp-255/0/0
                                     3 vcp-255/0/1
                                     3 vcp-255/0/2
                                     3 vcp-255/0/3
```

```

3 vcp-255/0/4
3 vcp-255/0/5
3 vcp-255/0/6
3 vcp-255/0/7

1 (FPC 1) Prsnt def456 ex4200-48p 128 Linecard 0 vcp-0
2 vcp-1

2 (FPC 2) Prsnt ghi789 ex4200-48p 128 Linecard 1 vcp-0
0 vcp-1

3 (FPC 3) Prsnt jkl012 ex4500-40f 255 Backup 4 vcp-0
5 vcp-1
0 vcp-255/0/0
0 vcp-255/0/1
0 vcp-255/0/2
0 vcp-255/0/3
0 vcp-255/0/4
0 vcp-255/0/5
0 vcp-255/0/6
0 vcp-255/0/7

4 (FPC 4) Prsnt mno345 ex4200-48p 128 Linecard 3 vcp-0
5 vcp-1

5 (FPC 5) Prsnt pqr678 ex4200-48p 128 Linecard 4 vcp-0
3 vcp-1

```

## Meaning

The `show virtual-chassis` command lists the member switches interconnected as a Virtual Chassis configuration with the member IDs that have been assigned by the primary and the roles. It also displays the neighbor members with which each member is interconnected.

## Troubleshooting

### IN THIS SECTION

- [Troubleshooting Nonoperational VCPs | 168](#)

To troubleshoot a Virtual Chassis configuration that is interconnected across wiring closets, perform these tasks:

### Troubleshooting Nonoperational VCPs

#### Problem

A user-configured VCP shows a status of **down**.

#### Solution

- Check the cable to make sure that it is properly and securely connected to the ports.
- Make sure the VCP that it has been explicitly set as an uplink VCP.
- Make sure that you have specified the options (*pic-slot*, *port*, and *member*) correctly.

### RELATED DOCUMENTATION

[Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) | 293](#)

## Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File

### IN THIS SECTION

- [Requirements | 169](#)
- [Overview and Topology | 170](#)
- [Configuration | 175](#)
- [Verification | 178](#)
- [Troubleshooting | 182](#)

You can deterministically control both the role and the member ID assigned to each member switch in an EX4200 Virtual Chassis configuration by creating a preprovisioned configuration file.

A preprovisioned configuration file links the serial number of each EX4200 switch in the configuration to a specified member ID and role. The serial number must be specified in the configuration file for the member to be recognized as part of the Virtual Chassis configuration.

**NOTE:** When you use a preprovisioned configuration, you cannot modify the primary-role priority or member ID of member switches through the user interfaces.

**NOTE:** After you have created a preprovisioned Virtual Chassis configuration, you can use the autoprovisioning feature to add member switches to that configuration. See *Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)*.

This example describes how to configure a Virtual Chassis across multiple wiring closets using a preprovisioned configuration file:

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.0 or later for EX Series switches
- Five EX4200-48P switches
- Five EX4200-24T switches
- Four XFP uplink modules

Before you create the preprovisioned configuration of the Virtual Chassis and interconnect the members across the wiring closets, be sure you have:

1. Made a list of the serial numbers of all the switches to be connected as a Virtual Chassis configuration.
2. Noted the desired role (**routing-engine** or **line-card**) of each switch. If you configure the member with a **routing-engine** role, it is eligible to function as a primary or backup. If you configure the member with a **line-card** role, it is not eligible to become a primary or backup.
3. Installed an uplink module in each of the member switches that will be interconnected across wiring closets. See *Installing an Uplink Module in an EX4200 Switch*.
4. Interconnected the member switches within each wiring closet using the dedicated VCPs on the rear panel of switches. See *Connecting a Virtual Chassis Cable to an EX4200 Switch*.
5. Powered on the switch that you plan to use as the primary switch (SWA-0).

6. Run the EZSetup program on SWA-0, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.

SWA-0 is going to be configured in the example to function as the primary of the Virtual Chassis configuration. Thus, the properties that you specify for SWA-0 will apply to the entire Virtual Chassis configuration, including all the member switches that you specify in the preprovisioned configuration file.

7. Configured SWA-0 with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis configuration, if desired.

```
[edit]
user@SWA-0# set interfaces vme unit 0 family inet address /ip-address/
mask/
```

## Overview and Topology

### IN THIS SECTION

- [Topology | 172](#)

You must select two members that you want to make eligible for election as primary of the Virtual Chassis configuration. When you list these two members in the preprovisioned configuration file, you designate both members as **routing-engine**. One will function as the primary of the Virtual Chassis configuration and the other will function as the backup.

You designate additional members, which are not eligible for election as primary, as having the **line-card** role in the preprovisioned configuration file.

In this example, five EX4200 switches (SWA-0 through SWA-4) are interconnected with their dedicated VCPs in wiring closet A and five EX4200 switches (SWA-5 through SWA-9) are interconnected with their dedicated VCPs in wiring closet B.

SWA-0 (in wiring closet A) is going to be the primary of the Virtual Chassis configuration. This example shows how to create a preprovisioned configuration file on SWA-0 for all member switches that will be interconnected in the Virtual Chassis configuration. The preprovisioned configuration file includes member IDs for the members in wiring closet A and for the members in wiring closet B.

SWA-5 (in wiring closet B) is going to be the backup of the Virtual Chassis configuration. Both SWA-0 and SWA-5 are specified in the preprovisioned configuration file with the role of **routing-engine**. All other members are specified with the role of **line-card**.

If all member switches could be interconnected with their dedicated VCPs, you could simply power on the switches after saving and committing the preprovisioned configuration file. The primary detects the connection of the members through the dedicated VCPs and applies the parameters specified in the preprovisioned configuration file.

However, the Virtual Chassis cables that interconnect the VCPs of member switches within a single wiring closet are not long enough to connect member switches across wiring closets. Instead, you can configure the uplink module ports or the SFP network ports on EX4200-24F switches as VCPs to interconnect the member switches in wiring closet A to the member switch in wiring closet B. For redundancy, this example connects uplink VCPs from two member switches in wiring closet A (SWA-0 and SWA-2) to two member switches (SWA-5 and SWA-7) in wiring closet B.

**NOTE:** You can use interfaces on SFP, SFP+, and XFP uplink modules and the SFP network ports on EX4200-24F switches as VCPs. When an uplink module port or SFP network port is set as a VCP, it cannot be used for any other purpose. The SFP uplink module has four 1-Gbps ports; the SFP+ uplink module has four 1-Gbps or two 10-Gbps ports; the XFP uplink module has two 10-Gbps ports. The uplink module ports that are not set as VCPs can be configured as trunk ports to connect to a distribution switch.

Because this particular preprovisioned configuration is for a Virtual Chassis that is interconnected across wiring closets, we will bring up the Virtual Chassis configuration in stages. First, we power on SWA-0 (without powering on any other switches) and create the preprovisioned configuration file. Then we power on the remaining switches in wiring closet A. If we check the status of the Virtual Chassis configuration at this point by using the `show virtual-chassis` command, it will display only **member 0** through **member 4**. The members that have not yet been interconnected will not be listed.

Next power on SWA-5 without powering on the remaining switches (SWA-6 through SWA-9) in wiring closet B. Bring up SWA-5 as a standalone switch and set one of its uplinks as a VCP prior to interconnecting it with the Virtual Chassis configuration in wiring closet A. Without this setting, SWA-5 cannot be detected as a member switch by the primary of the Virtual Chassis configuration.

You can set the uplink VCP of SWA-5 without running the EZSetup program by directly connecting to the console port. If you wish, you can run the EZSetup program and specify identification parameters. When you interconnect SWA-5 with the primary of the Virtual Chassis configuration, the primary overwrites any conflicting parameters.

After setting the VCP in SWA-5, connect this VCP with the VCP of SWA-0 in wiring closet A. SWA-5 (serial number pqr678) is specified as a **routing-engine** in the preprovisioned configuration file.

This example uses SWA-5 as the backup of the Virtual Chassis configuration. If a problem occurred in wiring closet A, SWA-5 would take control of the Virtual Chassis configuration and maintain the network connections. Specify both SWA-0 and SWA-5 as **routing-engine**. Because SWA-0 is powered



on prior to SWA-5, it has additional prioritization properties that cause it to be elected as primary of the Virtual Chassis configuration.

After being physically interconnected with SWA-0, SWA-5 reboots and comes up as **member 5** and as the backup of the Virtual Chassis configuration.

Power on the remaining switches (SWA-6 through SWA-9) in wiring closet B. The primary can now detect that all members are present. Finally, for redundancy, configure an additional VCP on SWA-7 through the primary.

## Topology

The topology for this example consists of:

- Three EX4200-48P switches (SWA-0, SWA-2, and SWA-4) in wiring closet A.
- Two EX4200-48P switches (SWA-5 and SWA-9) in wiring closet B.
- Two EX4200-24T switches (SWA-1 and SWA-3) in wiring closet A.
- Three EX4200-24T switches (SWA-6, SWA-7, and SWA-8) in wiring closet B.
- Four XFP uplink modules. Two are installed in wiring closet A and two are installed in wiring closet B.

[Table 21 on page 172](#) shows the Virtual Chassis configuration settings for a preprovisioned Virtual Chassis composed of member switches in different wiring closets.

**Table 21: Components of a Preprovisioned EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets**

Switch	Serial number	Member ID	Role	Uplink Module Ports	Hardware	Location
SWA-0	abc123	0	routing-engine	<b>xe-0/1/0</b>	EX4200-48P and XFP uplink module	Wiring closet A
SWA-1	def456	1	linecard		EX4200-24T	Wiring closet A
SWA-2	ghi789	2	linecard	<b>xe-2/1/0</b>	EX4200-48P and XFP uplink module	Wiring closet A

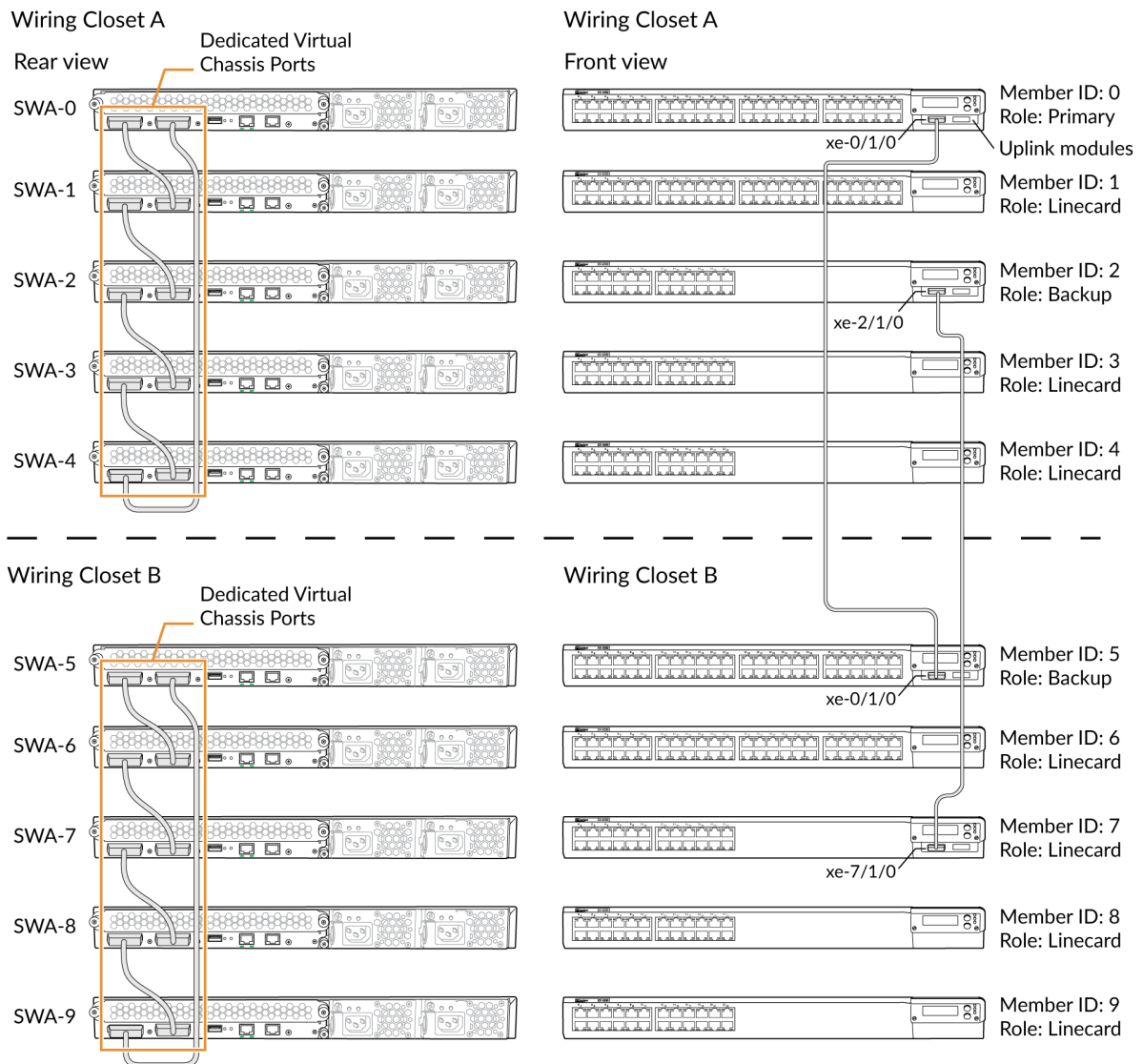
**Table 21: Components of a Preprovisioned EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets (Continued)**

Switch	Serial number	Member ID	Role	Uplink Module Ports	Hardware	Location
SWA-3	jkl012	3	linecard		EX4200-24T	Wiring closet A
SWA-4	mno345	4	linecard		EX4200-48P	Wiring closet A
SWA-5	pqr678	5	routing-engine	<b>xe-0/1/0</b>  <b>NOTE:</b> The member ID of SWA-5 is <b>0</b> at the time that its uplink module port is configured as a VCP.	EX4200-48P and XFP uplink module	Wiring closet B
SWA-6	stu901	6	linecard		EX4200-24T	Wiring closet B
SWA-7	vwx234	7	linecard	<b>xe-7/1/0</b>	EX4200-24T and XFP uplink module	Wiring closet B
SWA-8	ywa567	8	linecard		EX4200-24T	Wiring closet B
SWA-9	bcd890	9	linecard		EX4200-48P	Wiring closet B

Figure 16 on page 174 shows the different types of interconnections used for this Virtual Chassis configuration. The rear view shows that the member switches within each wiring closet are interconnected to each other using the dedicated VCPs. The front view shows that the uplink module ports that have been set as VCPs and interconnected across the wiring closets. The uplink module ports that are not set as VCPs can be configured as trunk ports to connect to a distribution switch.

**NOTE:** The interconnections shown in [Figure 16 on page 174](#) are the same as they would be for a configuration that was not preprovisioned across wiring closets.

**Figure 16: Maximum Size EX4200 Virtual Chassis Interconnected Across Wiring Closets**



## Configuration

### IN THIS SECTION

- [Procedure | 175](#)

To configure the Virtual Chassis across multiple wiring closets using a preprovisioned configuration, perform this task:

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis configuration.

### Procedure

#### Step-by-Step Procedure

To create a preprovisioned configuration for the Virtual Chassis:

1. Specify the preprovisioned configuration mode:

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

2. Specify all the members that will be included in the Virtual Chassis configuration, listing each switch's serial number with the desired member ID and the desired role:

```
[edit virtual-chassis]
user@SWA-0# set member 0 serial-number abc123 role routing-engine
user@SWA-0# set member 1 serial-number def456 role line-card
user@SWA-0# set member 2 serial-number ghi789 role line-card
user@SWA-0# set member 3 serial-number jkl012 role line-card
user@SWA-0# set member 4 serial-number mno345 role line-card
user@SWA-0# set member 5 serial-number pqr678 role routing-engine
user@SWA-0# set member 6 serial-number stu901 role line-card
user@SWA-0# set member 7 serial-number vwx234 role line-card
```

```
user@SWA-0# set member 8 serial-number yza567 role line-card
user@SWA-0# set member 9 serial-number bcd890 role line-card
```

3. Power on the member switches in wiring closet A.
4. Prepare the members in wiring closet A for interconnecting with the member switches in wiring closet B by setting uplink VCPs for member 0 and member 2:

```
user@SWA-0> request virtual-chassis vc-port (Dedicated VCP) set pic-slot 1 port 0
user@SWA-2> request virtual-chassis vc-port set pic-slot 1 port 0 member 2
```

**NOTE:**

- For redundancy, this example sets an uplink VCP in both SWA-0 and SWA-2.
- This example omits the specification of the member **0** in setting the uplink for SWA-0. The command applies by default to the switch where it is executed.

5. Power on SWA-5 and connect to it. This switch comes up as member ID **0** and functions as primary of itself. Although SWA-5 is listed in the preprovisioned configuration file, it is not a present member of the Virtual Chassis configuration that has been powered on thus far. In order for the primary to detect SWA-5 as a connected member, you must first set an uplink VCP on SWA-5 and interconnect that VCP with the uplink VCP of SWA-0.
6. Set the first uplink of SWA-5 to function as a VCP. Because SWA-5 has been powered on as a separate switch and is still operating independently at this point, its member ID is **0**.

```
user@SWA-5> request virtual-chassis vc-port set pic-slot 1 port 0
```

**NOTE:** This example omits the specification of the member **0** in configuring the uplink for SWA-5 (at this point the member ID of SWA-5 is still **0**). The command applies by default to the switch where it is executed.

7. Power off SWA-5 and connect the fiber cable from SWA-5 uplink VCP **xe-0/1/0** to the uplink VCP **xe-0/1/0** on SWA-0.
8. Power on SWA-5.

9. Now that SWA-5 has been brought up as **member 5** of the Virtual Chassis configuration, power on the remaining switches (SWA-6 through SWA-9) in wiring closet B. They are interconnected with SWA-5 using the dedicated VCPs on the rear panel and are therefore detected by the primary as interconnected members. If you check the status of the Virtual Chassis configuration at this point, all the members that were specified in the preprovisioned configuration file should be displayed as present. Additional configuration for member switches can now be done through the primary switch.
10. Set one uplink module port of SWA-7 to function as a VCP:

```
user@SWA-0> request virtual-chassis vc-port set pic-slot 1 port 0
member 7
```

## Results

Display the results of the configuration on SWA-0:

```
[edit]
user@SWA-0# show
virtual-chassis {
  member 0 {
    role routing-engine;
    serial-number abc123;
  }
  member 1 {
    role line-card;
    serial-number def456;
  }
  member 2 {
    role line-card;
    serial-number ghi789;
  }
  member 3 {
    role line-card;
    serial-number jkl012;
  }
  member 4 {
    role line-card;
    serial-number mno345;
  }
  member 5 {
```

```
        role routing-engine;
        serial-number pqr678;
    }
    member 6 {
        role line-card;
        serial-number stu901;
    }
    member 7 {
        role line-card;
        serial-number vwx234;
    }
    member 8 {
        role line-card;
        serial-number yza567;
    }
    member 9 {
        role line-card;
        serial-number bcd890;
    }
    preprovisioned;
}
```

## Verification

### IN THIS SECTION

- [Verifying the Member IDs and Roles of the Member Switches | 178](#)
- [Verifying That the Dedicated VCPs and Uplink VCPs Are Operational | 180](#)

To confirm that the configuration is working properly, perform these tasks:

### Verifying the Member IDs and Roles of the Member Switches

#### Purpose

Verify that the member IDs and roles are all set as expected.

## Action

Display the members of the Virtual Chassis configuration:

```
user@SWA-0> show virtual-chassis
```

```
Preprovisioned Virtual Chassis
```

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

Member ID	Status	Serial No	Model	Mastership		Neighbor List	
				Priority	Role	ID	Interface
0 (FPC 0)	Prsnt	abc123	ex4200-48p	129	Master*	1	vcp-0
						4	vcp-1
						5	1/0
1 (FPC 1)	Prsnt	def456	ex4200-24t	0	Linecard	2	vcp-0
						0	vcp-1
2 (FPC 2)	Prsnt	ghi789	ex4200-48p	0	Linecard	3	vcp-0
						1	vcp-1
						7	1/0
3 (FPC 3)	Prsnt	jkl012	ex4200-24t	0	Linecard	4	vcp-0
						2	vcp-1
4 (FPC 4)	Prsnt	mno345	ex4200-48p	0	Linecard	0	vcp-0
						3	vcp-1
5 (FPC 5)	Prsnt	pqr678	ex4200-48p	129	Backup	6	vcp-0
						9	vcp-1
						0	1/0
6 (FPC 6)	Prsnt	stu901	ex4200-24t	0	Linecard	7	vcp-0
						5	vcp-1
7 (FPC 7)	Prsnt	vwxyz34	ex4200-24t	0	Linecard	8	vcp-0
						6	vcp-1
						2	1/0
8 (FPC 8)	Prsnt	zya567	ex4200-24t	0	Linecard	9	vcp-0
						7	vcp-1



9 (FPC 9)	Prsnt	bc7890	ex4200-48p	0	Linecard	5 vcp-0
						8 vcp-1

Meaning

The output shows that all members listed in the preprovisioned configuration file are connected to the Virtual Chassis configuration. It confirms that SWA-0 (member 0) is functioning as the primary of the Virtual Chassis configuration. The other switch configured with the **routing-engine** role (SWA-5) is functioning as the backup. The **Neighbor List** displays the interconnections of the member VCPs.

Verifying That the Dedicated VCPs and Uplink VCPs Are Operational

Purpose

Verify that the dedicated VCPs interconnecting the member switches within each wiring closet and the uplink module VCPs interconnecting the member switches across wiring closets are operational.

Action

Display the VCP interfaces:

```
user@SWA-0> show virtual-chassis vc-port all-members

fpc0:
-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up
1/0            Configured Up

fpc1:
-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up

fpc2:
```

```

-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up
1/0            Configured Up

```

fpc3:

```

-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up

```

fpc4:

```

-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up

```

fpc5:

```

-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up
1/0            Configured Up

```

fpc6:

```

-----
Interface      Type      Status  Speed      Neighbor
or              (mbps)    ID      Interface
PIC / Port
vcp-0          Dedicated Up
vcp-1          Dedicated Up

```

fpc7:

```

-----
```

Interface or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID    Interface
vcp-0	Dedicated	Up		
vcp-1	Dedicated	Up		
1/0	Configured	Up		

fpc8:

Interface or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID    Interface
vcp-0	Dedicated	Up		
vcp-1	Dedicated	Up		

fpc9:

Interface or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID    Interface
vcp-0	Dedicated	Up		
vcp-1	Dedicated	Up		

Meaning

The dedicated VCPs interconnecting the member switches within wiring closets are displayed as **vcp-0** and **vcp-1**. The uplink module VCPs interconnecting member switches (members 0, 2, 5, and 7) across wiring closets are displayed as **1/0** and **1/1** and identified as **Configured**.

Troubleshooting

IN THIS SECTION

- [Troubleshooting Nonoperational VCPs | 183](#)

To troubleshoot a preprovisioned Virtual Chassis configuration that is interconnected across wiring closets, perform these tasks:

### Troubleshooting Nonoperational VCPs

#### Problem

A VCP shows a status of **down**.

#### Solution

Check the cable to make sure that it is properly and securely connected to the ports.

### RELATED DOCUMENTATION

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

[Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets | 148](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

[Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)

## Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis

### IN THIS SECTION

- [Requirements | 184](#)
- [Overview and Topology | 184](#)
- [Configuration | 186](#)
- [Verification | 188](#)
- [Troubleshooting | 190](#)

You can deterministically control both the role and the member ID assigned to each member switch in a mixed EX4200 and EX4500 Virtual Chassis configuration by creating a preprovisioned configuration.

**NOTE:** When you use a preprovisioned configuration, you cannot modify the primary-role priority or member ID of member switches through the user interfaces.

This example describes how to configure a mixed EX4200 and EX4500 Virtual Chassis using a preprovisioned configuration file:

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later for EX Series switches

**NOTE:** You must use Junos OS Release 11.4 or later if you are including three or more EX4500 switches in an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis.

- Two EX4500 switches, each with a Virtual Chassis module
- Two EX4200 switches

Before you create the preprovisioned configuration of the Virtual Chassis and interconnect the members across the wiring closets, be sure you have:

1. Made a list of the serial numbers of all the switches to be connected as a Virtual Chassis configuration.
2. Noted the desired role (**routing-engine** or **line-card**) for each switch.
3. Ensured that the same version of Junos OS is running on all current or prospective member switches.

## Overview and Topology

### IN THIS SECTION

- [Topology | 185](#)

A preprovisioned configuration file links the serial number of each switch in the Virtual Chassis configuration to a specified member ID and role. The serial number must be specified in the configuration file for the member to be recognized as part of the Virtual Chassis configuration.

You designate additional members, which are not eligible for election as primary, as having the **line-card** role in the preprovisioned configuration file.

In this example, two EX4500 switches (SWA-0 and SWA-1) are in the routing-engine role and interconnected to two EX4200 switches (SWA-2 and SWA-3) in the linecard role using dedicated Virtual Chassis ports (VCPs).

SWA-0 will be the primary of the Virtual Chassis configuration. This example shows how to create a preprovisioned configuration file on SWA-0 for all member switches that will be interconnected in the Virtual Chassis configuration. The preprovisioned configuration file includes member IDs for all member switches.

SWA-1 will be the backup of the Virtual Chassis configuration. Both SWA-0 and SWA-1 are specified in the preprovisioned configuration file with the role of **routing-engine**. All other members are specified with the role of **line-card**.

After all member switches are interconnected with their dedicated VCPs, you can simply power on the switches after saving and committing the preprovisioned configuration file. The primary detects the connection of the members through the dedicated VCPs and applies the parameters specified in the preprovisioned configuration file.

**NOTE:** You can use interfaces on SFP, SFP+, and XFP uplink modules as VCPs. When an uplink module port or SFP network port is set as a VCP, it cannot be used for any other purpose.

We will bring up the Virtual Chassis configuration in stages. First, we power on SWA-0 (without powering on any other switches) and create the preprovisioned configuration file. Then we power on the remaining switches.

## Topology

The topology for this example consists of:

- Two EX4500 switches (SWA-0 and SWA-1)
- Two EX4200 switches (SWA-2 and SWA-3)

[Table 22 on page 186](#) shows the Virtual Chassis configuration settings for a preprovisioned Virtual Chassis.

Table 22: Components of a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis

Switch	Serial number	Member ID	Role	Hardware
SWA-0	abc123	0	routing-engine	EX4500 switch with a Virtual Chassis module
SWA-1	def456	1	routing-engine	EX4500 switch with a Virtual Chassis module
SWA-2	ghi789	2	line-card	EX4200 switch
SWA-3	jkl012	3	line-card	EX4200 switch

## Configuration

### IN THIS SECTION

- [Procedure | 186](#)

To configure the Virtual Chassis across multiple wiring closets using a preprovisioned configuration, perform this task:

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis configuration.

## Procedure

### Step-by-Step Procedure

To create a preprovisioned configuration for the Virtual Chassis:

1. Power on the EX4500 switch (SWA-0) in the primary role..

2. Set the PIC mode to Virtual Chassis mode on SWA-0:

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

3. Set the Virtual Chassis mode to mixed:

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

4. Reboot the switch.
5. After the switch reboots, specify the preprovisioned configuration mode:

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

6. Specify all members to be included in the Virtual Chassis configuration, listing each switch's serial number with the desired member ID and the desired role:

```
[edit virtual-chassis]
user@SWA-0# set member 0 serial-number abc123 role routing-engine
user@SWA-0# set member 1 serial-number def456 role line-card
user@SWA-0# set member 2 serial-number ghi789 role line-card
user@SWA-0# set member 3 serial-number jkl012 role line-card
```

7. Power on the remaining switches.
8. Set the Virtual Chassis mode to mixed on the remaining switches:

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

9. Reboot the switches.
10. When the reboot completes, physically cable the switches together using the dedicated VCPs.



## Results

Display the results of the configuration on SWA-0:

```
[edit]
user@SWA-0# show
virtual-chassis {
  member 0 {
    role routing-engine;
    serial-number abc123;
  }
  member 1 {
    role line-card;
    serial-number def456;
  }
  member 2 {
    role line-card;
    serial-number ghi789;
  }
  member 3 {
    role line-card;
    serial-number jkl012;
  }
  preprovisioned;
}
```

## Verification

### IN THIS SECTION

- [Verifying the Member IDs and Roles of the Member Switches | 189](#)

To confirm that the configuration is working properly, perform these tasks:

## Verifying the Member IDs and Roles of the Member Switches

### Purpose

Verify that the member IDs and roles are all set as expected.

### Action

Display the members of the Virtual Chassis configuration:

```
user@SWA-0> show virtual-chassis
Preprovisioned Virtual Chassis
Virtual Chassis ID: 0000.e255.0000
```

Member ID	Status	Serial No	Model	Mastership		Neighbor List	
				Priority	Role	ID	Interface
0 (FPC 0)	Prsnt	abc123	ex4500-40f	129	Master*	1	vcp-1
						3	vcp-0
1 (FPC 1)	Prsnt	def456	ex4500-40f	0	Backup	0	vcp-0
						2	vcp-1
2 (FPC 2)	Prsnt	ghi789	ex4200-48p	0	Linecard	1	vcp-0
						3	vcp-1
3 (FPC 3)	Prsnt	jkl012	ex4200-24t	0	Linecard	2	vcp-0
						0	vcp-1

### Meaning

The output shows that all members listed in the preprovisioned configuration file are connected to the Virtual Chassis configuration. It confirms that SWA-0 (member **0**) is functioning as the primary of the Virtual Chassis configuration. The other switch configured with the **routing-engine** role (SWA-1) is functioning as the backup. The **Neighbor List** displays the interconnections of the member VCPs.

## Troubleshooting

### IN THIS SECTION

- [Troubleshooting Nonoperational VCPs | 190](#)

To troubleshoot a preprovisioned Virtual Chassis configuration that is interconnected across wiring closets, perform these tasks:

### Troubleshooting Nonoperational VCPs

#### Problem

A VCP shows a status of **down**.

#### Solution

Check the cable to make sure that it is properly and securely connected to the ports.

### RELATED DOCUMENTATION

---

[Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) | 269](#)

---

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

---

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

## Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch

### IN THIS SECTION

- [Requirements | 191](#)
- [Overview and Topology | 192](#)
- [Configuration | 194](#)
- [Verification | 198](#)
- [Troubleshooting | 199](#)

EX Series switches allow you to combine multiple Ethernet links into one logical interface for higher bandwidth and redundancy. The ports that are combined in this manner are referred to as a link aggregation group (LAG) or bundle. The number of Ethernet links you can combine into a LAG depends on your EX Series switch model.

This example describes how to configure uplink LAGs to connect a Virtual Chassis access switch to a Virtual Chassis distribution switch:

### Requirements

This example uses the following software and hardware components:

- Junos OS Release 9.0 or later for EX Series switches
- Two EX4200-48P switches
- Two EX4200-24F switches
- Four XFP uplink modules

Before you configure the LAGs, be sure you have:

- Configured the Virtual Chassis switches. See ["Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)" on page 245](#).
- Configured the uplink ports on the switches as trunk ports. See [Configuring Gigabit Ethernet Interfaces \(CLI Procedure\)](#).

## Overview and Topology

For maximum speed and resiliency, you can combine uplinks between an access switch and a distribution switch into LAGs. Using LAGs can be particularly effective when connecting a multimember Virtual Chassis access switch to a multimember Virtual Chassis distribution switch.

The Virtual Chassis access switch in this example is composed of two member switches. Each member switch has an uplink module with two 10-Gigabit Ethernet ports. These ports are configured as trunk ports, connecting the access switch with the distribution switch.

Configuring the uplinks as LAGs has the following advantages:

- Link Aggregation Control Protocol (LACP) can optionally be configured for link negotiation.
- It doubles the speed of each uplink from 10 Gbps to 20 Gbps.
- If one physical port is lost for any reason (a cable is unplugged or a switch port fails, or one member switch is unavailable), the logical port transparently continues to function over the remaining physical port.

The topology used in this example consists of one Virtual Chassis access switch and one Virtual Chassis distribution switch. The access switch is composed of two EX4200-48P switches (SWA-0 and SWA-1), interconnected to each other with their Virtual Chassis ports (VCPs) as member switches of Host-A. The distribution switch is composed of two EX4200-24F switches (SWD-0 and SWD-1), interconnected with their VCPs as member switches of Host-D.

Each member of the access switch has an uplink module installed. Each uplink module has two ports. The uplinks are configured to act as trunk ports, connecting the access switch with the distribution switch. One uplink port from SWA-0 and one uplink port from SWA-1 are combined as LAG ae0 to SWD-0. This link is used for one VLAN. The remaining uplink ports from SWA-0 and from SWA-1 are combined as a second LAG connection (ae1) to SWD-1. LAG ae1 is used for another VLAN.

**NOTE:** If the remote end of the LAG link is a security device, LACP might not be supported because security devices require a deterministic configuration. In this case, do not configure

LACP. All links in the LAG are permanently operational unless the switch detects a link failure within the Ethernet physical layer or data link layers.

Figure 17: Topology for LAGs Connecting an EX4200 Virtual Chassis Access Switch to an EX4200 Virtual Chassis Distribution Switch

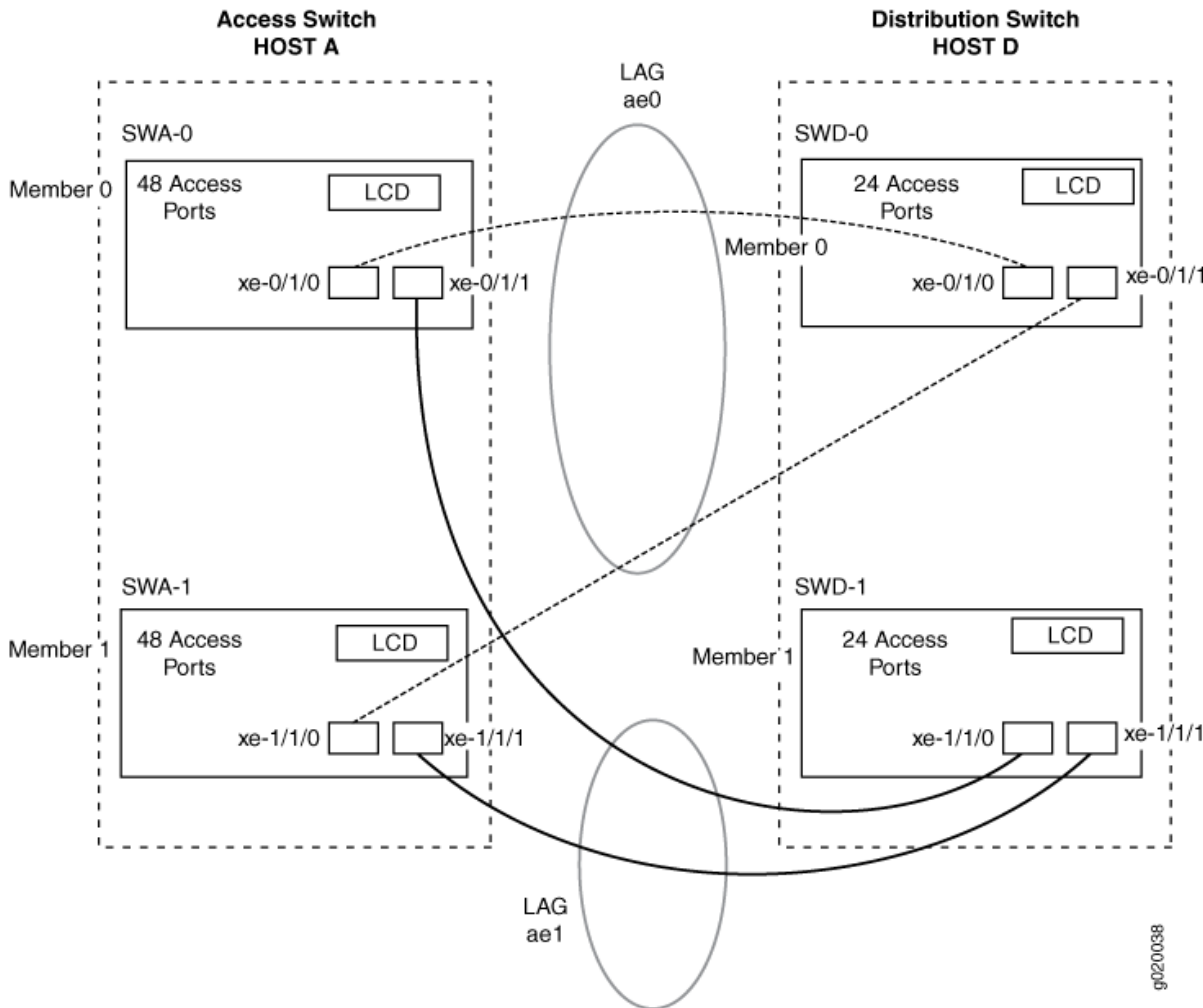


Table 23 on page 194 details the topology used in this configuration example.

**Table 23: Components of the Topology for Connecting a Virtual Chassis Access Switch to a Virtual Chassis Distribution Switch**

Switch	Hostname and VCID	Base Hardware	Uplink Module	Member ID	Trunk Port
SWA-0	Host-A Access switch VCID 1	EX4200-48P switch	One XFP uplink module	0	xe-0/1/0 to SWD-0  xe-0/1/1 to SWD-1
SWA-1	Host-A Access switch VCID 1	EX4200-48P switch	One XFP uplink module	1	xe-1/1/0 to SWD-0  xe-1/1/1 to SWD-1
SWD-0	Host-D Distribution switch VCID 4	EX4200 L-24F switch	One XFP uplink module	0	xe-0/1/0 to SWA-0  xe-0/1/1 to SWA-1
SWD-1	Host-D Distribution switch VCID 4	EX4200 L-24F switch	One XFP uplink module	1	xe-1/1/0 to SWA-0  xe-1/1/1 to SWA-1

## Configuration

### IN THIS SECTION

- Procedure | 195
- Results | 197

To configure two uplink LAGs from the Virtual Chassis access switch to the Virtual Chassis distribution switch.

## Procedure

### CLI Quick Configuration

To quickly configure aggregated Ethernet high-speed uplinks between a Virtual Chassis access switch and a Virtual Chassis distribution switch, copy the following commands and paste them into the switch terminal window:

```
[edit]

set chassis aggregated-devices ethernet device-count 2
set interfaces ae0 aggregated-ether-options minimum-links 1
set interfaces ae0 aggregated-ether-options link-speed 10g
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options link-speed 10g
set interfaces ae0 unit 0 family inet address 192.0.2.0/25
set interfaces ae1 unit 0 family inet address 192.0.2.128/25
set interfaces xe-0/1/0 ether-options 802.3ad ae0
set interfaces xe-1/1/0 ether-options 802.3ad ae0
set interfaces xe-0/1/1 ether-options 802.3ad ae1
set interfaces xe-1/1/1 ether-options 802.3ad ae1
```

### Step-by-Step Procedure

To configure aggregated Ethernet high-speed uplinks between a Virtual Chassis access switch and a Virtual Chassis distribution switch:

1. Specify the number of LAGs to be created on the chassis:

```
[edit chassis]
user@Host-A# set aggregated-devices ethernet device-count 2
```

2. Specify the number of links that need to be present for the ae0 LAG interface to be up:

```
[edit interfaces]
user@Host-A# set ae0 aggregated-ether-options minimum-links 1
```



3. Specify the number of links that need to be present for the ae1 LAG interface to be up:

```
[edit interfaces]
user@Host-A# set ae1 aggregated-ether-options minimum-links 1
```

4. Specify the media speed of the ae0 link:

```
[edit interfaces]
user@Host-A# set ae0 aggregated-ether-options link-speed 10g
```

5. Specify the media speed of the ae1 link:

```
[edit interfaces]
user@Host-A# set ae1 aggregated-ether-options link-speed 10g
```

6. Specify the interface ID of the uplinks to be included in LAG ae0:

```
[edit interfaces]
user@Host-A# set xe-0/1/0 ether-options 802.3ad ae0
user@Host-A# set xe-1/1/0 ether-options 802.3ad ae0
```

7. Specify the interface ID of the uplinks to be included in LAG ae1:

```
[edit interfaces]
user@Host-A# set xe-0/1/1 ether-options 802.3ad ae1
user@Host-A# set xe-1/1/1 ether-options 802.3ad ae1
```

8. Specify that LAG ae0 belongs to the subnet for the employee broadcast domain:

```
[edit interfaces]
user@Host-A# set ae0 unit 0 family inet address 192.0.2.0/25
```

9. Specify that LAG ae1 belongs to the subnet for the guest broadcast domain:

```
[edit interfaces]
user@Host-A# set ae1 unit 0 family inet address 192.0.2.128/25
```

## Results

Display the results of the configuration:

```
[edit]
chassis {
  aggregated-devices {
    ethernet {
      device-count 2;
    }
  }
}
interfaces {
  ae0 {
    aggregated-ether-options {
      link-speed 10g;
      minimum-links 1;
    }
    unit 0 {
      family inet {
        address 192.0.2.0/25;
      }
    }
  }
  ae1 {
    aggregated-ether-options {
      link-speed 10g;
      minimum-links 1;
    }
    unit 0 {
      family inet {
        address 192.0.2.128/25;
      }
    }
  }
  xe-0/1/0 {
```

```
        ether-options {  
            802.3ad ae0;  
        }  
    }  
    xe-1/1/0 {  
        ether-options {  
            802.3ad ae0;  
        }  
    }  
    xe-0/1/1 {  
        ether-options {  
            802.3ad ae1;  
        }  
    }  
    xe-1/1/1 {  
        ether-options {  
            802.3ad ae1;  
        }  
    }  
}
```

## Verification

### IN THIS SECTION

- [Verifying That LAG ae0 Has Been Created | 198](#)
- [Verifying That LAG ae1 Has Been Created | 199](#)

To verify that switching is operational and two LAGs have been created, perform these tasks:

### Verifying That LAG ae0 Has Been Created

#### Purpose

Verify that LAG ae0 has been created on the switch.

Action

show interfaces ae0 terse

Interface	Admin	Link	Proto	Local	Remote
ae0	up	up			
ae0.0	up	up	inet	192.0.2.0/25	

Meaning

The output confirms that the ae0 link is up and shows the family and IP address assigned to this link.

Verifying That LAG ae1 Has Been Created

Purpose

Verify that LAG ae1 has been created on the switch

Action

show interfaces ae1 terse

Interface	Admin	Link	Proto	Local	Remote
ae1	up	down			
ae1.0	up	down	inet	192.0.2.128/25	

Meaning

The output shows that the ae1 link is down.

Troubleshooting

IN THIS SECTION

- [Troubleshooting a LAG That Is Down | 200](#)

## Troubleshooting a LAG That Is Down

### Problem

The `show interfaces terse` command shows that the LAG is down

### Solution

Check the following:

- Verify that there is no configuration mismatch.
- Verify that all member ports are up.
- Verify that a LAG is part of family ethernet switching (Layer 2 LAG) or family inet (Layer 3 LAG).
- Verify that the LAG member is connected to the correct LAG at the other end.
- Verify that the LAG members belong to the same switch (or the same Virtual Chassis).

### RELATED DOCUMENTATION

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

[Example: Connecting an EX Series Access Switch to a Distribution Switch](#)

*Virtual Chassis Cabling Configuration Examples for EX4200 Switches*

*Installing an Uplink Module in an EX4200 Switch*

## Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch

### IN THIS SECTION

- [Requirements | 201](#)
- [Overview and Topology | 201](#)
- [Configuring LACP for the LAGs on the Virtual Chassis Access Switch | 202](#)

- [Configuring LACP for the LAGs on the Virtual Chassis Distribution Switch | 203](#)
- [Verification | 205](#)
- [Troubleshooting | 207](#)

EX Series switches allow you to combine multiple Ethernet links into one logical interface for higher bandwidth and redundancy. The ports that are combined in this manner are referred to as a link aggregation group (LAG) or bundle. EX Series switches allow you to further enhance these links by configuring Link Aggregation Control Protocol (LACP).

This example describes how to overlay LACP on the LAG configurations that were created in ["Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch" on page 191](#):

## Requirements

This example uses the following software and hardware components:

- Junos OS Release 9.0 or later for EX Series switches
- Two EX4200-48P switches
- Two EX4200-24F switches
- Four EX Series XFP uplink modules

Before you configure LACP, be sure you have:

- Set up the Virtual Chassis switches. See ["Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)" on page 245](#).
- Configured the uplink ports on the switches as trunk ports. See [Configuring Gigabit Ethernet Interfaces \(CLI Procedure\)](#).
- Configured the LAGs. See ["Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch" on page 191](#).

## Overview and Topology

This example assumes that you are familiar with ["Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch" on page 191](#). The topology in this example is exactly the same as the topology in that other example. This example shows how to use LACP to enhance the LAG functionality.

LACP exchanges are made between *actors* (the transmitting link) and *partners* (the receiving link). The LACP mode can be either active or passive.

**NOTE:** If the actor and partner are both in passive mode, they do not exchange LACP packets, which results in the aggregated Ethernet links not coming up. By default, LACP is in passive mode. To initiate transmission of LACP packets and responses to LACP packets, you must enable LACP in active mode.

By default, the actor and partner send LACP packets every second.

The interval can be fast (every second) or slow (every 30 seconds).

## Configuring LACP for the LAGs on the Virtual Chassis Access Switch

### IN THIS SECTION

- [Procedure](#) | 202

To configure LACP for the access switch LAGs, perform these tasks.

### Procedure

#### CLI Quick Configuration

To quickly configure LACP for the access switch LAGs, copy the following commands and paste them into the switch terminal window:

```
[edit]
set interfaces ae0 aggregated-ether-options lacp active periodic fast
                        set interfaces ae1 aggregated-ether-options lacp active periodic
fast
```

### Step-by-Step Procedure

To configure LACP for Host-A LAGs ae0 and ae1:

1. Specify the aggregated Ethernet options for both bundles:

```
[edit interfaces]
user@Host-A#set ae0 aggregated-ether-options lacp active periodic fast
user@Host-A#set ae1 aggregated-ether-options lacp active periodic fast
```

## Results

Display the results of the configuration:

```
[edit interfaces]
user@Host-A# show
ae0 {
  aggregated-ether-options {
    lacp {
      active;
      periodic fast;
    }
  }
}
ae1 {
  aggregated-ether-options {
    lacp {
      active;
      periodic fast;
    }
  }
}
```

## Configuring LACP for the LAGs on the Virtual Chassis Distribution Switch

### IN THIS SECTION

- [Procedure](#) | 204

To configure LACP for the two uplink LAGs from the Virtual Chassis access switch to the Virtual Chassis distribution switch, perform these tasks.



## Procedure

### CLI Quick Configuration

To quickly configure LACP for the distribution switch LAGs, copy the following commands and paste them into the switch terminal window:

```
[edit interfaces]
set ae0 aggregated-ether-options lacp passive periodic fast
set ae1 aggregated-ether-options lacp passive periodic
fast
```

### Step-by-Step Procedure

To configure LACP for Host D LAGs ae0 and ae1:

1. Specify the aggregated Ethernet options for both bundles:

```
[edit interfaces]
user@Host-D#set ae0 aggregated-ether-options lacp passive periodic fast
user@Host-D#set ae1 aggregated-ether-options lacp passive periodic fast
```

## Results

Display the results of the configuration:

```
[edit interfaces]
user@Host-D# show
ae0 {
    aggregated-ether-options {
        lacp {
            passive;
            periodic fast;
        }
    }
}
ae1 {
    aggregated-ether-options {
        lacp {
```

```
        passive
        periodic fast;
    }
}
}
```

Verification

IN THIS SECTION

[Verifying the LACP Settings | 205](#)

[Verifying That the LACP Packets Are Being Exchanged | 206](#)

To verify that LACP packets are being exchanged, perform these tasks:

Verifying the LACP Settings

Purpose

Verify that LACP has been set up correctly.

Action

Use the `show lacp interfaces interface-name` command to check that LACP has been enabled as active on one end.

```
user@Host-A> show lacp interfaces xe-0/1/0

Aggregated interface: ae0

LACP state:      Role  Exp  Def  Dist  Col  Syn  Aggr  Timeout  Activity

  xe-0/1/0      Actor  No  Yes   No   No   No   Yes    Fast    Active

  xe-0/1/0      Partner No  Yes   No   No   No   Yes    Fast    Passive

LACP protocol:  Receive State  Transmit State      Mux State
```

xe-0/1/0	Defaulted	Fast periodic	Detached
----------	-----------	---------------	----------

Meaning

The output indicates that LACP has been set up correctly and is active at one end.

Verifying That the LACP Packets Are Being Exchanged

Purpose

Verify that LACP packets are being exchanged.

Action

Use the `show interfaces aex statistics` command to display LACP information.

```
user@Host-A> show interfaces ae0 statistics

Physical interface: ae0, Enabled, Physical link is Down
  Interface index: 153, SNMP ifIndex: 30
  Link-level type: Ethernet, MTU: 1514, Speed: Unspecified, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Disabled, Minimum links needed: 1,
  Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Current address: 02:19:e2:50:45:e0, Hardware address: 02:19:e2:50:45:e0
  Last flapped   : Never
  Statistics last cleared: Never
    Input packets : 0
    Output packets: 0
  Input errors: 0, Output errors: 0

Logical interface ae0.0 (Index 71) (SNMP ifIndex 34)
  Flags: Hardware-Down Device-Down SNMP-Traps Encapsulation: ENET2
  Statistics      Packets      pps      Bytes      bps
  Bundle:
    Input :          0          0          0          0
    Output:          0          0          0          0
  Protocol inet
```

```
Flags: None
Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
Destination: 10.10.10/24, Local: 10.10.10.1, Broadcast: 10.10.10.255
```

## Meaning

The output here shows that the link is down and that no protocol data units (PDUs) are being exchanged.

## Troubleshooting

### IN THIS SECTION

- [Troubleshooting a Nonworking LACP Link | 207](#)

To troubleshoot a nonworking LACP link, perform these tasks:

### Troubleshooting a Nonworking LACP Link

#### Problem

The LACP link is not working.

#### Solution

Check the following:

- Remove the LACP configuration and verify whether the static LAG is up.
- Verify that LACP is configured at both ends.
- Verify that LACP is not passive at both ends.
- Verify whether LACP protocol data units (PDUs) are being exchanged by running the `monitor traffic-interface lag-member detail` command.

## RELATED DOCUMENTATION

[Example: Connecting an EX Series Access Switch to a Distribution Switch](#)

[Virtual Chassis Cabling Configuration Examples for EX4200 Switches](#)

[Installing an Uplink Module in an EX4200 Switch](#)

## Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails

### IN THIS SECTION

- [Requirements | 208](#)
- [Overview and Topology | 209](#)
- [Configuration | 211](#)
- [Verification | 212](#)

The Virtual Chassis fast failover feature is a hardware-assisted failover mechanism that automatically reroutes traffic and reduces traffic loss in the event of a link or switch failure. If a link between two members fails, traffic flow between those members must be rerouted quickly so that there is minimal traffic loss.

Fast failover is enabled by default on all dedicated EX4200 Virtual Chassis ports (VCPs).

This example describes how to configure fast failover on uplink module VCPs in an EX4200 Virtual Chassis configuration:

### Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.3 or later for EX Series switches
- Six EX4200-24T switches
- Four SFP uplink modules

Before you begin configuring fast failover, be sure you have:

1. Mounted the switches. See *Mounting an EX4200 Switch*.

2. Cabled the switches in a multiple-ring topology to create the Virtual Chassis configuration. See *Connecting a Virtual Chassis Cable to an EX4200 Switch*.

## Overview and Topology

### IN THIS SECTION

- [Topology | 210](#)

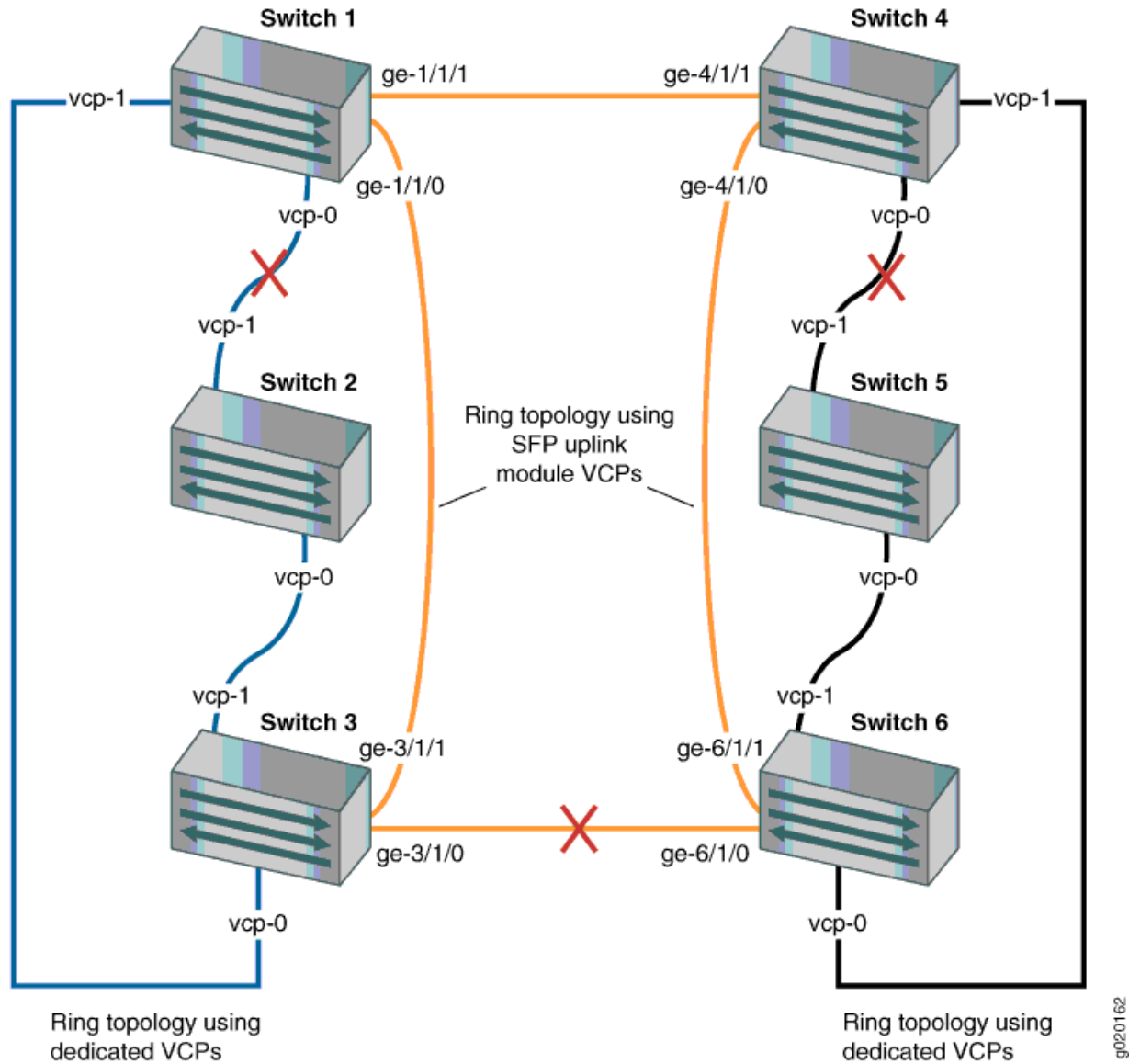
In a Virtual Chassis configuration, fast failover automatically reroutes traffic and reduces traffic loss in the event of a link failure or a member switch failure. By default, fast failover is enabled on all dedicated VCPs. If you configure uplink module ports as VCPs, you must manually configure fast failover on these ports.

For fast failover to be effective, the Virtual Chassis members must be configured in a ring topology. The ring topology can be formed by using either dedicated VCPs or user-configured uplink module VCPs. Fast failover is supported only in a ring topology that uses identical port types, for example, either a topology that uses all dedicated VCPs or one that uses all uplink module VCPs. Fast failover is not supported in a ring topology that includes both dedicated VCPs and uplink module VCPs. Fast failover is supported, however, in a Virtual Chassis configuration that consists of multiple rings.

This example shows how to enable fast failover on uplink module VCPs.

Figure 18 on page 210 shows an example of a multiple-ring topology.

Figure 18: Traffic Redirected by Fast Failover After VCP Link Failures in a Topology with Multiple Rings



### Topology

The topology for this example consists of six switches:

- Six EX4200-24T switches, four of which have an SFP uplink module installed (switches 1, 3, 4, and 6)

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 211](#)
- [Procedure | 211](#)

To configure the fast failover feature on uplink module VCPs:

### CLI Quick Configuration

To configure fast failover on all SFP uplink module VCPs, copy the following command and paste it into the terminal window on switch 1:

```
[edit]  
set virtual-chassis fast-failover ge
```

### Procedure

#### Step-by-Step Procedure

To configure fast failover on SFP uplink module VCPs:

1. Enable fast failover on all SFP uplink module VCPs in the Virtual Chassis configuration:

```
[edit]  
user@switch1# set virtual-chassis fast-failover ge
```

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis.



## Results

Check the results of the configuration:

```
[edit virtual-chassis]
user@switch1# show
fast-failover {
    ge;
}
```

## Verification

### IN THIS SECTION

- [Verifying That Fast Failover Is Enabled | 212](#)

To confirm that fast failover is enabled on SFP uplink module VCPs in the Virtual Chassis configuration, perform this task:

### Verifying That Fast Failover Is Enabled

#### Purpose

Verify that fast failover has been enabled in a Virtual Chassis configuration.

#### Action

Issue the `show virtual-chassis fast-failover` command.

```
user@switch1> show virtual-chassis fast-failover
```

```
Fast failover on dedicated VCP ports: Enabled
Fast failover on XE uplink VCP ports: Disabled
Fast failover on GE uplink VCP ports: Enabled
```

## Meaning

Fast failover is enabled on all dedicated VCPs and SFP uplink module VCPs in the Virtual Chassis configuration.

## RELATED DOCUMENTATION

[Configuring Fast Failover in an EX Series Virtual Chassis | 297](#)

[Disabling Fast Failover in an EX Series Virtual Chassis | 298](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

[Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)

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

### IN THIS SECTION

- [Requirements | 214](#)
- [Overview and Topology | 214](#)
- [Configuration | 215](#)
- [Verification | 216](#)

There are two scenarios in which separate Virtual Chassis merge:

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

You can explicitly assign a Virtual Chassis ID (VCID) so that, when two EX4200 Virtual Chassis configurations merge, the ID that you assigned takes precedence over the automatically assigned VCIDs and becomes the ID of the newly merged Virtual Chassis configuration.

This example shows how to assign the VCID on an EX4200 Virtual Chassis. This process is identical on an EX4500 Virtual Chassis and on a mixed EX4200 and EX4500 Virtual Chassis.

This example describes how to assign the VCID in an EX4200 Virtual Chassis configuration:

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.3 or later for EX Series switches
- Two EX4200-48P switches
- Two EX4200-24T switches

Before you begin, be sure you have:

1. Installed the switches. See *Mounting an EX4200 Switch*.
2. Cabled the switches to create the Virtual Chassis configuration. See *Connecting a Virtual Chassis Cable to an EX4200 Switch*.

## Overview and Topology

### IN THIS SECTION

- [Topology | 214](#)

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

### Topology

The topology for this example consists of four switches:

- Two EX4200-24T switches
- Two EX4200-48P switches

The switches are connected as a four-member Virtual Chassis configuration and are identified as switch-A, switch-B, switch-C, and switch-D. The primary is switch-A.

## Configuration

### IN THIS SECTION

- [Procedure](#) | 215

To assign the Virtual Chassis ID in a Virtual Chassis configuration:

### Procedure

#### CLI Quick Configuration

To assign a Virtual Chassis ID so that when two Virtual Chassis configurations merge, the ID that you assigned takes precedence over the automatically assigned Virtual Chassis IDs and becomes the ID of the newly merged Virtual Chassis configuration, copy the following command and paste it into the terminal window:

```
[edit]  
set virtual-chassis id 9622.6ac8.5345
```

#### Step-by-Step Procedure

To assign the Virtual Chassis ID in a Virtual Chassis configuration:

1. Assign the Virtual Chassis ID:

```
[edit]  
user@switch-A# set virtual-chassis id 9622.6ac8.5345
```

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis configuration.

## Verification

### IN THIS SECTION

- [Verifying That the Virtual Chassis ID Is Assigned | 216](#)

To verify that the Virtual Chassis ID has been assigned as you intended, perform these tasks:

### Verifying That the Virtual Chassis ID Is Assigned

#### Purpose

Verify that the Virtual Chassis ID has been assigned in a Virtual Chassis configuration.

#### Action

Issue the `show configuration virtual-chassis id` command.

```
user@switch-A> show configuration virtual-chassis id  
id 9622.6ac8.5345;
```

#### Meaning

The Virtual Chassis ID has been assigned as 9622.6ac8.5345.

## RELATED DOCUMENTATION

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

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

[Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)

## Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports

### IN THIS SECTION

- Requirements | 217
- Overview and Topology | 218
- Configuration | 220
- Verification | 224
- Troubleshooting | 228

You can form link aggregation groups (LAGs) between EX4200 Virtual Chassis member switches in different wiring closets using uplink Virtual Chassis ports (VCPs) and, on EX4200-24F switches, network VCPs. LAGs balance traffic across the member links, increase the uplink bandwidth, and provide increased availability. To form LAGs using uplink or network VCPs, you configure the uplink module interfaces or network interfaces on the member switches as VCPs and connect the VCPs using fiber-optic cables. For the LAGs to form, the uplink or network VCPs on each member switch that will form a LAG must operate at the same link speed and you must interconnect at least two uplink or network VCPs on each of those members. You can connect uplink or network VCPs operating at different link speeds, but they will not form a LAG.

**NOTE:** The LAGs formed by VCPs are different from LAGs formed by Virtual Chassis network interfaces. For more information on LAGs formed by network interfaces, see *Understanding Virtual Chassis Port Link Aggregation*.

This example shows how to configure uplink module interfaces and network interfaces as VCPs on multiple member switches of a Virtual Chassis configuration and then connect them to form LAGs:

### Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.6 or later for EX Series switches
- Five EX4200 switches, one of which is an EX4200-24F model
- Two SFP uplink modules

- Two XFP uplink modules

Before you configure the uplink module interfaces and network interfaces on Virtual Chassis member switches as VCPs and interconnect the members to form a LAG, be sure you have:

1. Installed the SFP uplink modules in the SWA-0 and SWA-2 switches and installed the XFP uplink modules in the SWA-1 and SWA-3 switches. See *Installing an Uplink Module in an EX4200 Switch*.
2. Powered on SWA-0, connected it to the network, and run the EZSetup program. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.
3. Configured SWA-0 with the virtual management Ethernet (VME) interface for remote, out-of-band management of the Virtual Chassis configuration, if desired. See "[Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\)](#)" on page 295.
4. Ensured that SWA-1 is not powered on and then interconnected SWA-0 and SWA-1 using the dedicated VCPs on the rear panel.

**NOTE:** The interfaces for the dedicated VCPs are operational by default. They do not need to be configured.

5. Ensured that SWA-2, SWA-3, and SWA-4 are not powered on. They are not connected in any way, so when they are initially powered on they will be standalone switches.

## Overview and Topology

### IN THIS SECTION

- [Topology](#) | 218

In this example, five EX4200 switches will be interconnected to form LAGs for ease of monitoring and manageability. Two of these switches (SWA-0 and SWA-1) are located in wiring closet A and the three others (SWA-2, SWA-3, and SWA-4) are located in wiring closet B. SWA-0 will form one LAG with SWA-2 and another LAG with SWA-4, and SWA-1 will form a LAG with SWA-3.

### Topology

We will use fiber-optic cables connected to the uplink and network VCPs to interconnect the member switches in wiring closet A to the member switches in wiring closet B.

We will specify the highest primary-role priority value (255) for SWA-0 to make it the primary before we power on SWA-1. Because SWA-0 and SWA-1 are interconnected with the dedicated VCPs, the primary detects that SWA-1 is a member of its Virtual Chassis configuration and assigns it a member ID.

We will use SWA-2 as the backup of the Virtual Chassis configuration. We will configure the same primary-role priority value for SWA-2 (255) that we configured for the primary. Because we power on SWA-0 before we power on SWA-2, SWA-0 retains primary role of the Virtual Chassis configuration.

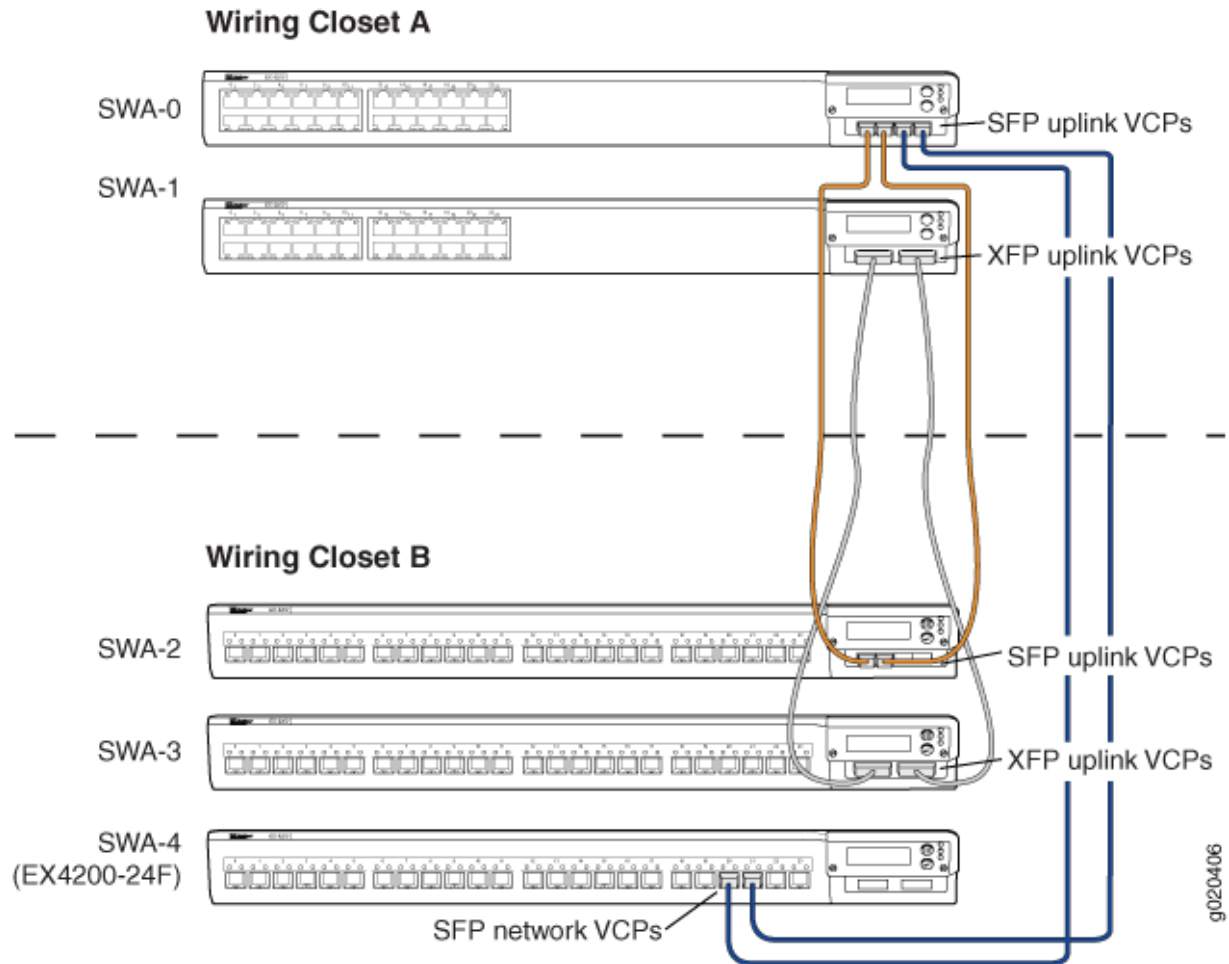
**NOTE:** We recommend setting identical primary-role priority values for the primary and backup members for high availability and smooth transition of primary role in case the original primary becomes unavailable.

We will configure the uplink module interfaces on three of the switches as uplink VCPs. On the EX4200-24F switch we will configure two of the network interfaces as VCPs. We will interconnect two of the SFP uplink VCPs on SWA-0 with two of the SFP uplink VCPs on SWA-2. Similarly, we will interconnect the two XFP uplink VCPs on SWA-1 with the two XFP uplink VCPs on SWA-3. Finally, we will connect the two remaining SFP uplink VCPs on SWA-0 with two network VCPs on SWA-4. As a result, three LAGs will be automatically formed.



Figure 19 on page 220 shows the interconnections used to form LAGs using uplink VCPs and the network VCPs after the procedure below has been completed.

Figure 19: EX4200 Virtual Chassis Interconnected Across Wiring Closets to Form LAGs



## Configuration

### IN THIS SECTION

- Procedure | 221

To configure the Virtual Chassis uplink module interfaces and network interfaces as uplink VCPs and interconnect them between two wiring closets to form LAGs, perform this task:

## Procedure

### Step-by-Step Procedure

To configure a Virtual Chassis across multiple wiring closets and interconnect them to form LAGs:

1. Configure the primary-role priority of SWA-0 (member 0) to be the highest possible value (255), thereby ensuring that it functions as the primary of the expanded Virtual Chassis configuration:

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

2. Power on SWA-1.
3. Prepare the members in wiring closet A for interconnecting with the member switches in wiring closet B by setting all of the SFP uplink module interfaces on SWA-0 and two of the uplink module interfaces on SWA-1 as uplink VCPs:

```
user@SWA-0> request virtual-chassis vc-port set pic-slot 1 port 0
user@SWA-0> request virtual-chassis vc-port set pic-slot 1 port 1
user@SWA-0> request virtual-chassis vc-port set pic-slot 1 port 2
user@SWA-0> request virtual-chassis vc-port set pic-slot 1 port 3
user@SWA-0> request virtual-chassis vc-port set pic-slot 1 port 0 member 1
user@SWA-0> request virtual-chassis vc-port set pic-slot 1 port 1 member
1
```

**NOTE:** This example omits the specification of the **member *member-id*** option in configuring the uplink VCPs for SWA-0 (and, later, for SWA-2). The command applies by default to the switch where it is executed.

4. Power on SWA-2.
5. If SWA-2 was previously configured, revert it to the factory default configuration. See *Reverting to the Default Factory Configuration for the EX Series Switch*.
6. Prepare SWA-2 in wiring closet B by configuring its primary-role priority to be the highest possible value (255). Its member ID is currently 0, because it is not yet interconnected with the other

members of the Virtual Chassis configuration. It is operating as a standalone switch. Its member ID will change when it is interconnected.

```
[edit virtual-chassis]
user@SWA-2# set member 0 mastership-priority 255
```

**NOTE:** SWA-2 is configured with the same primary-role priority value that we configured for SWA-0. However, the longer uptime of SWA-0 ensures that, once the interconnection is made, SWA-0 functions as the primary and SWA-2 functions as the backup.

7. Specify two of the SFP uplink module interfaces in SWA-2 as uplink VCPs. The member IDs are **0**, because they are not yet interconnected with the other members of the Virtual Chassis configuration:

**NOTE:** The settings of the uplink VCPs remain intact when SWA-2 reboots and joins the Virtual Chassis configuration as member 2.

```
user@SWA-2> request virtual-chassis vc-port set pic-slot 1 port 0
user@SWA-2> request virtual-chassis vc-port set pic-slot 1 port 1
```

8. Power off SWA-2.
9. Physically interconnect SWA-0 and SWA-2 across wiring closets using two of the uplink VCPs on each switch.
10. Power on SWA-2. SWA-2 joins the Virtual Chassis configuration and a LAG is automatically formed between SWA-0 and SWA-2. In addition, although SWA-0 and SWA-2 have the same primary-role priority value (255), SWA-0 was powered on first and thus has longer uptime. This results in SWA-0 retaining primary role while SWA-2 reboots and joins the now expanded Virtual Chassis configuration as the backup, with member ID **2**.
11. Power on SWA-3.
12. If SWA-3 was previously configured, revert it to the factory default configuration.

13. Specify both XFP uplink module interfaces in SWA-3 as uplink VCPs:

```
user@SWA-3> request virtual-chassis vc-port set pic-slot 1 port 0
user@SWA-3> request virtual-chassis vc-port set pic-slot 1 port 1
```

14. Power off SWA-3.
15. Physically interconnect SWA-3 with SWA-2 using their dedicated VCPs.
16. Physically interconnect SWA-1 and SWA-3 across wiring closets using their uplink VCPs.
17. Power on SWA-3. It joins the Virtual Chassis configuration as member **3**.

**NOTE:** Member ID **3** is assigned to SWA-3 because SWA-3 was powered on after members 0, 1, and 2.

A LAG is automatically formed between SWA-1 and SWA-3. In addition, both SWA-1 and SWA-3 have the default primary-role priority value (128) and function in a linecard role.

18. Power on SWA-4.
19. If SWA-4 was previously configured, revert it to the factory default configuration.
20. Configure two of the network interfaces on SWA-4 as uplink VCPs:

```
user@SWA-4> request virtual-chassis vc-port set pic-slot 0 port 20
user@SWA-4> request virtual-chassis vc-port set pic-slot 0 port 21
```

21. Power off SWA-4.
22. Physically interconnect SWA-4 and SWA-0 across wiring closets using the network VCPs on SWA-4 and the two remaining SFP uplink VCPs on SWA-0.
23. Power on SWA-4. A LAG is automatically formed between SWA-4 and SWA-0. In addition, SWA-4 joins the Virtual Chassis configuration in the linecard role.

## Results

Display the results of the configuration on SWA-0:

```
user@SWA-0> show configuration virtual-chassis
  member 0 {
    mastership-priority 255;
  }
  member 1 {
    mastership-priority 128;
  }
  member 2 {
    mastership-priority 255;
  }
  member 3 {
    mastership-priority 128;
  }
  member 4 {
    mastership-priority 128;
  }
}
```

## Verification

### IN THIS SECTION

- [Verifying the Member IDs and Roles of the Member Switches | 224](#)
- [Verifying That the VCPs Are Operational | 226](#)

To confirm that the configuration is working properly, perform these tasks:

### Verifying the Member IDs and Roles of the Member Switches

#### Purpose

Verify that all the interconnected member switches are included within the Virtual Chassis configuration and that their roles are assigned appropriately.

Action

Display the members of the Virtual Chassis configuration:

```
user@SWA-0> show virtual-chassis

Virtual Chassis ID: 0000.e255.00e0

Member ID  Status  Serial No  Model      Mastership  Priority  Role  Neighbor List
              ID              ID              ID              ID              ID              ID
0 (FPC 0)  Prsnt   abc123    ex4200-48p  255        Master*   1 vcp-0
              1 vcp-1
              2 vcp-255/1/0
              2 vcp-255/1/1
              4 vcp-255/0/20
              4 vcp-255/0/21

1 (FPC 1)  Prsnt   def456    ex4200-24t  128        Linecard  0 vcp-0
              0 vcp-1
              3 vcp-255/1/0
              3 vcp-255/1/1

2 (FPC 2)  Prsnt   ghi789    ex4200-48p  255        Backup   3 vcp-0
              3 vcp-1
              0 vcp-255/1/0
              0 vcp-255/1/1

3 (FPC 3)  Prsnt   jkl012    ex4200-24t  128        Linecard  2 vcp-0
              2 vcp-1
              1 vcp-255/1/0
              1 vcp-255/1/1

4 (FPC 4)  Prsnt   mno345    ex4200-24f  128        Linecard  0 vcp-255/1/2
              0 vcp-255/1/3
```

### Meaning

The `show virtual-chassis` command lists the member switches interconnected in a Virtual Chassis configuration with the member IDs that have been assigned by the primary, the primary-role priority values, and the roles. It also displays the neighbor members with which each member is interconnected by the dedicated VCPs, by uplink VCPs, and by network VCPs.

### Verifying That the VCPs Are Operational

#### Purpose

Verify that the dedicated VCPs interconnecting member switches in wiring closets A and B and the uplink and network VCPs interconnecting the member switches between wiring closets are operational.

#### Action

Display the Virtual Chassis interfaces:

```

user@SWA-0> show virtual-chassis vc-port all-members

fpc0:
-----
Interface  Type      Trunk  Status  Speed  Neighbor
or          ID              (mbps)  ID  Interface
PIC / Port
vcp-0      Dedicated  1      Up      32000  1      vcp-0
vcp-1      Dedicated  2      Up      32000  1      vcp-1
1/0        Configured 3      Up      1000   2      vcp-255/1/0
1/1        Configured 3      Up      1000   2      vcp-255/1/1
1/2        Configured 4      Up      1000   4      vcp-255/0/20
1/3        Configured 4      Up      1000   4      vcp-255/0/21

fpc1:
-----
Interface  Type      Trunk  Status  Speed  Neighbor
or          ID              (mbps)  ID  Interface
PIC / Port
vcp-0      Dedicated  1      Up      32000  0      vcp-0
vcp-1      Dedicated  2      Up      32000  0      vcp-1
1/0        Configured 3      Up      10000  3      vcp-255/1/0
1/1        Configured 3      Up      10000  3      vcp-255/1/1

```

fpc2:

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

fpc3:

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

fpc4:

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

## Meaning

The dedicated VCPs are displayed as **vcp-0** and **vcp-1**. The uplink module interfaces that have been set as uplink VCPs are displayed as **1/0**, **1/1**, **1/2**, and **1/3**. The network interfaces that have been set as VCPs are displayed as **0/20** and **0/21**. The neighbor interface names of uplink and network VCPs are of the form **vcp-255/pic/port**—for example, **vcp-255/1/0**. In that name, **vcp-255** indicates that the interface is a VCP, **1** is the uplink PIC number, and **0** is the port number. The **fpc** number is the same as



the member ID. The trunk ID is a positive number ID assigned to the LAG formed by the Virtual Chassis. If no LAG is formed, the value is **-1**.

**NOTE:** Each switch assigns the trunk IDs to its local interfaces. As a result, the pair of interfaces that form one end of a LAG on one switch will have the same trunk ID, and the pair of interfaces that form the other end of the LAG will have the same trunk ID, but the trunk IDs on either end of the LAG might be different. For example, in [Figure 19 on page 220](#), the uplink VCPs **1/2** and **1/3** on SWA-0 form a LAG with the network VCPs **0/20** and **0/21** on SWA-4. Uplink VCPs **1/2** and **1/3** on SWA-0 both have trunk ID **4**, while network VCPs **0/20** and **0/21** on SWA-4 both have trunk ID **3**. The trunk IDs are different between the switches because SWA-0 assigns the trunk IDs for its local uplink VCPs and SWA-4 assigns the trunk IDs for its local VCPs.

## Troubleshooting

### IN THIS SECTION

- [Troubleshooting Nonoperational VCPs | 228](#)

To troubleshoot a Virtual Chassis configuration that is interconnected across wiring closets, perform this task:

### Troubleshooting Nonoperational VCPs

#### Problem

An uplink VCP shows a status of **down**.

#### Solution

- Check the cable to make sure that it is properly and securely connected to the interfaces.
- If the VCP is an uplink module interface, make sure that it has been explicitly set as an uplink VCP.
- If the VCP is an uplink module interface, make sure that you have specified the options (*pic-slot*, *port*, and *member*) correctly.

## RELATED DOCUMENTATION

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

[Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet | 119](#)

[Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration | 140](#)

*Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*

## Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches

### IN THIS SECTION

- [Requirements | 229](#)
- [Overview and Topology | 230](#)
- [Configuration | 231](#)
- [Verification | 232](#)

The automatic software update feature automatically updates the Junos OS version on prospective member switches as they are added to a Virtual Chassis configuration of EX4200 switches so the new member switch immediately joins the EX4200 Virtual Chassis configuration and is put in the active state. If the software version on the new switch is not the same as the version running on the primary, the primary keeps the new switch in the inactive state. If you have not enabled the automatic software update feature, you will have to manually install the correct software version on each prospective member switch as it is added to the Virtual Chassis configuration.

This example describes how to configure the Virtual Chassis automatic software update feature:

### Requirements

This example uses the following hardware and software components:

- Three EX4200 switches
- Junos OS Release 10.0 or later for EX Series switches

Before you begin, be sure you have:

1. Ensured that two member switches are running the same version of Junos OS for EX Series switches so that they can form the initial Virtual Chassis configuration.
2. Cabled and powered on those two switches to create the Virtual Chassis configuration. See *Connecting a Virtual Chassis Cable to an EX4200 Switch*.
3. Ensured that you know the name or the URL of the software package to be used by the automatic software update feature.
4. If you are going to perform an automatic software update, ensure that the version of Junos OS running on the Virtual Chassis is compatible with the version of Junos OS running on the prospective member switch for automatic software update. See *Understanding Automatic Software Update on Virtual Chassis Member Switches*.

## Overview and Topology

### IN THIS SECTION

- [Topology | 230](#)

### Topology

For a standalone EX4200 switch to join an existing Virtual Chassis configuration, it must be running the same version of Junos OS that is running on the Virtual Chassis primary. If the software version on the new switch is not the same as the version running on the primary, the primary keeps the new switch in the inactive state.

The topology for this example consists of three EX Series switches. Two of the switches are connected in a Virtual Chassis configuration and are therefore running the same version of Junos OS for EX Series switches. The third switch is a standalone switch that is running a different software version than the Virtual Chassis member switches. In this example, we will enable the automatic software update feature on the Virtual Chassis configuration and then add the third switch to the configuration. The primary will detect the presence of the new switch, check the software version running on the new switch, and, because it is not the same version currently running on the primary, will update the software version on the new switch and reboot the switch so that it can join the Virtual Chassis configuration and immediately be put in the active state.

## Configuration

### IN THIS SECTION

- [Procedure](#) | [231](#)

To configure automatic software update, perform this task:

### Procedure

#### Step-by-Step Procedure

To configure automatic software update:

1. Enable automatic software update and configure the path to the software package:

```
[edit]
user@switch# set virtual-chassis auto-sw-update package-name /var/tmp/jinstall-
ex-4200-10.0R1.1-domestic-signed.tgz
```

2. Connect the new switch to the existing Virtual Chassis configuration, and power on the switch.

### Results

Check the results of the configuration:

```
[edit virtual-chassis]
user@switch# show
auto-sw-update {
    package-name /var/tmp/jinstall-ex-4200-10.0R1.1-domestic-signed.tgz;
}
```



## Meaning

Because in the initial two-member Virtual Chassis configuration member 0 was the primary and member 1 was the backup, the output shows that the new switch has been assigned member ID 2 and has been given the Linecard role. The Status field shows that member 2 is Prsnt, which means that it is in the active state.

## RELATED DOCUMENTATION

---

*Configuring Automatic Software Update on Virtual Chassis Member Switches*

*Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)*

# Configuration Tasks

## IN THIS CHAPTER

- [Configuring an EX2200 Virtual Chassis \(CLI Procedure\) | 235](#)
- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) | 240](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)
- [Installing Software on a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Switches \(CLI Procedure\) | 259](#)
- [Adding a New Switch to an Existing EX3300 Virtual Chassis \(CLI Procedure\) | 261](#)
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\) | 263](#)
- [Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) | 269](#)
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 271](#)
- [Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 273](#)
- [Removing or Replacing a Member Switch of a Virtual Chassis Configuration | 275](#)
- [Removing an EX4200, EX4500, or EX4550 Switch From a Mixed Virtual Chassis \(CLI Procedure\) | 283](#)
- [Configuring Primary Role of a Virtual Chassis | 284](#)
- [Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port | 287](#)
- [Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) | 293](#)
- [Setting an Uplink Port as a Virtual Chassis Port Using the LCD Panel | 294](#)
- [Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\) | 295](#)
- [Configuring the Timer for the Backup Member to Start Using Its Own MAC Address as Primary of a Virtual Chassis | 296](#)
- [Configuring Fast Failover in an EX Series Virtual Chassis | 297](#)
- [Disabling Fast Failover in an EX Series Virtual Chassis | 298](#)
- [Disabling Split and Merge in a Virtual Chassis | 299](#)

- [Configuring Automatic Software Update on Virtual Chassis Member Switches | 301](#)
- [Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis \(CLI Procedure\) | 303](#)
- [Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge | 308](#)
- [Configuring Graceful Routing Engine Switchover in a Virtual Chassis | 308](#)
- [Resetting the EX4500 Switch to Run as a Standalone Switch \(CLI Procedure\) | 309](#)
- [Enabling VLAN Pruning for Broadcast, Multicast, and Unknown Unicast Traffic in an EX Series Virtual Chassis \(CLI Procedure\) | 311](#)

## Configuring an EX2200 Virtual Chassis (CLI Procedure)

### IN THIS SECTION

- [Configuring an EX2200 Virtual Chassis with a Preprovisioned Configuration File | 236](#)
- [Configuring an EX2200 Virtual Chassis with a Nonprovisioned Configuration File | 238](#)

You configure an EX2200 Virtual Chassis by configuring interfaces connecting EX2200 switches into Virtual Chassis ports (VCPs). Optical interfaces on EX2200 and EX2200-C switches can be configured into VCPs and can be used to connect EX2200 switches into a Virtual Chassis over short and long distances (up to 49.7 miles). All RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T RJ-45 transceivers, on EX2200 and EX2200-C switches can also be configured into VCPs.

An EX2200 Virtual Chassis can be configured with either:

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



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

**NOTE:** You do not have to purchase a license to configure an EX2200 Virtual Chassis starting in Junos OS release 12.3R2.

Prior to Junos OS release 12.3R2, two enhanced feature licenses (EFLs)—one for the switch in the primary role and another for the switch in the backup role—must be purchased to create an EX2200 Virtual Chassis. See [Understanding Software Licenses for EX Series Switches](#).

This topic includes:

## Configuring an EX2200 Virtual Chassis with a Preprovisioned Configuration File

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

To configure a Virtual Chassis using a preprovisioned configuration:

**NOTE:** We recommend that you physically cable the ports as the final step of this procedure. You can, however, configure an EX2200 Virtual Chassis while the cables are physically connected.

1. Make a list of the serial numbers of all the switches to be connected in a Virtual Chassis configuration.
2. Note the desired role (`routing-engine` or `line-card`) of each switch. If you configure the member with a `routing-engine` role, it is eligible to function in the primary or backup role. If you configure the member with a `line-card` role, it is not eligible to function in the primary or backup role.
3. Power on only the switch that you plan to use as the primary switch.
4. Run the EZSetup program on the primary switch, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.

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

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

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

6. Specify the preprovisioned configuration mode:

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

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

```
[edit virtual-chassis]
user@switch# set member 0 serial-number abc123 role routing-engine
user@switch# set member 1 serial-number def456 role routing-engine
user@switch# set member 2 serial-number ghi789 role line-card
user@switch# set member 3 serial-number jkl012 role line-card
```

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

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

9. Power on the other member switches. The member IDs and roles have been determined by the configuration, so you can power on the member switches in any order.
10. On each individual member switch, configure the ports that will be used to interconnect the EX2200 member switches into VCPs.

You can configure a port on an EX2200 switch as a VCP using the following command:

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

where *pic-slot-number* is the PIC slot number. The PIC slot number is *0* when you are configuring a built-in port as a VCP and *1* when you are configuring an uplink port as a VCP.

For instance, if you wanted to set built-in port 20 as a VCP:

```
user@switch> request virtual-chassis vc-port set pic-slot 0 port 20
```

If you wanted to set uplink port 2 as a VCP:

```
user@switch> request virtual-chassis vc-port set pic-slot 1 port 2
```

The VCPs automatically bundle into a Link Aggregation Group when two or more interfaces are configured into VCPs between the same two member switches. See *Understanding Virtual Chassis Port Link Aggregation*.

**NOTE:** You cannot modify the primary-role priority when you are using a preprovisioned configuration. The primary-role priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file. The two Routing Engines are assigned the same primary-role priority value. However, the member that was powered on first has higher prioritization according to the primary-role election algorithm. See *Understanding How the Primary in a Virtual Chassis Is Elected*.

## Configuring an EX2200 Virtual Chassis with a Nonprovisioned Configuration File

You can use nonprovisioned configuration to configure an EX2200 Virtual Chassis.

To configure the Virtual Chassis using a nonprovisioned configuration:

**NOTE:** We recommend that you physically cable the ports as the final step of this procedure. You can, however, configure an EX2200 Virtual Chassis while the cables are physically connected.

1. Power on only the switch that you plan to use as the primary switch.
2. Run the EZSetup program on the primary switch, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.

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

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

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

4. (Optional) Configure primary-role priority for the other member switches:

```
[edit virtual-chassis]
user@switch# set member 0 mastership-priority 255
user@switch# set member 1 mastership-priority 255
```

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

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

6. Power on the other member switches.
7. On each individual member switch, configure the ports that will be used to interconnect the EX2200 member switches into VCPs.

You can configure a port on an EX2200 switch as a VCP using the following command:

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

where *pic-slot-number* is the PIC slot number. The PIC slot number is *0* when you are configuring a built-in port as a VCP and *1* when you are configuring an uplink port as a VCP.

For instance, if you wanted to set built-in port 20 as a VCP:

```
user@switch> request virtual-chassis vc-port set pic-slot 0 port 20
```

If you wanted to set uplink port 2 as a VCP:

```
user@switch> request virtual-chassis vc-port set pic-slot 1 port 2
```

The VCPs automatically bundle into a Link Aggregation Group when two or more interfaces are configured into VCPs between the same two member switches. See *Understanding Virtual Chassis Port Link Aggregation*.

**NOTE:** If you do not edit the Virtual Chassis configuration file, a nonprovisioned configuration is generated by default. The primary-role priority value for each member switch is 128. The primary role is selected by default. You can change the role that is performed by the members by modifying the primary-role priority. See *Configuring Primary Role of a Virtual Chassis*. We recommend that you specify the same primary-role priority value for the desired primary and backup members. In this example, the highest possible primary-role priority has been assigned to two members. However, the member that was powered on first has higher prioritization according to the primary-role election algorithm. See *Understanding How the Primary in a Virtual Chassis Is Elected*. The other members use the default primary-role priority in this example, which configures them to function in the role of linecard.

**NOTE:** If you want to change the member ID that the primary has assigned to a member switch, use the `request virtual-chassis renumber` command.

## RELATED DOCUMENTATION

*Virtual Chassis Overview for Switches*

*Understanding EX Series Virtual Chassis*

[Understanding EX2200 Virtual Chassis Hardware Configuration](#)

*Configuring Primary Role of a Virtual Chassis*

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

## Configuring an EX3300 Virtual Chassis (CLI Procedure)

### IN THIS SECTION

- [Configuring an EX3300 Virtual Chassis with a Preprovisioned Configuration File | 241](#)
- [Configuring an EX3300 Virtual Chassis with a Nonprovisioned Configuration File | 243](#)

You must interconnect EX3300 switches using uplink ports configured as Virtual Chassis ports (VCPs). Uplink ports 2 and 3 on an EX3300 switch are configured as VCPs by default. You do not have to configure these interfaces to connect EX3300 switches together in a Virtual Chassis.

The uplink ports configured as VCPs are used to connect EX3300 switches into a Virtual Chassis in the same or different wiring closets over short and long distances (up to 6.2 miles).

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

An EX3300 Virtual Chassis can be configured with either:

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

This topic includes:

## Configuring an EX3300 Virtual Chassis with a Preprovisioned Configuration File

Preprovisioning a Virtual Chassis configuration allows you to assign the member ID and role for each switch in the Virtual Chassis. Preprovisioning is supported for an EX3300 Virtual Chassis.

To configure a Virtual Chassis using a preprovisioned configuration:

1. Make a list of the serial numbers of all the switches to be connected in a Virtual Chassis configuration.
2. Note the desired role (**routing-engine** or **line-card**) of each switch. If you configure the member with a **routing-engine** role, it is eligible to function in the primary or backup role. If you configure the member with a **line-card** role, it is not eligible to function in the primary or backup role.
3. Interconnect the member switches using uplink ports 2 and 3 of your EX3300 switches. See [Virtual Chassis Cabling Configuration Examples for EX3300 Switches](#).

**NOTE:** For management purposes, we recommend arranging the switches in member ID sequence, either from top to bottom or from bottom to top (0–5).

4. Power on only the switch that you plan to use as the primary switch. Do not power on the other switches at this time.

5. Run the EZSetup program on the primary switch, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.

**NOTE:** The properties that you specify for the primary switch apply to the entire Virtual Chassis configuration, including all the members listed in the preprovisioned configuration file.

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

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

7. Specify the preprovisioned configuration mode:

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

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

```
[edit virtual-chassis]
user@switch# set member 0 serial-number abc123 role routing-engine
user@switch# set member 1 serial-number def456 role line-card
user@switch# set member 2 serial-number ghi789 role line-card
user@switch# set member 3 serial-number jkl012 role line-card
user@switch# set member 4 serial-number mno345 role line-card
user@switch# set member 5 serial-number pqr678 role routing-engine
```

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

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

10. Power on the member switches.

**NOTE:** You cannot modify the primary-role priority when you are using a preprovisioned configuration. The primary-role priority values are generated automatically and controlled by the

role that is assigned to the member switch in the configuration file. The two Routing Engines are assigned the same primary-role priority value. However, the member that was powered on first has higher prioritization according to the primary-role election algorithm. See *Understanding How the Primary in a Virtual Chassis Is Elected*.

## Configuring an EX3300 Virtual Chassis with a Nonprovisioned Configuration File

Nonprovisioned configuration can be used to configure an EX3300 Virtual Chassis.

To configure the Virtual Chassis using a nonprovisioned configuration:

1. Interconnect the member switches using uplink ports 2 and 3 on your EX3300 switches. See [Virtual Chassis Cabling Configuration Examples for EX3300 Switches](#).

**NOTE:** For management purposes, we recommend arranging the switches in member ID sequence, either from top to bottom or from bottom to top.

2. Power on only the switch that you plan to use as the primary switch (SWA-0). Do not power on the other switches at this time.
3. Run the EZSetup program on SWA-0, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.

**NOTE:** The properties that you specify for SWA-0 apply to the entire Virtual Chassis configuration, including all the members interconnected through VCPs.

4. (Optional) Configure SWA-0 with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis:

```
[edit]
user@SWA-0# set interfaces vme unit 0 family inet address /ip-address/mask/
```

5. (Optional) Configure primary-role priority for the primary, backup, and other members:

```
[edit virtual-chassis]
user@SWA-0# set member 0 mastership-priority 255
user@SWA-0# set member 3 mastership-priority 255
```



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

```
[edit virtual-chassis]
user@SWA-0# set no-split-detection
```

7. Power on the member switches in sequential order, one by one.

**NOTE:** If you do not edit the Virtual Chassis configuration file, a nonprovisioned configuration is generated by default. The primary-role priority value for each member switch is **128**. The primary role is selected by default. You can change the role that is performed by the members by modifying the primary-role priority. See *Configuring Primary Role of a Virtual Chassis*. We recommend that you specify the same primary-role priority value for the desired primary and backup members. In this example, the highest possible primary-role priority has been assigned to two members. However, the member that was powered on first has higher prioritization according to the primary-role election algorithm. See *Understanding How the Primary in a Virtual Chassis Is Elected*. The other members use the default primary-role priority in this example, which configures them to function in the role of linecard.

**NOTE:** If you want to change the member ID that the primary has assigned to a member switch, use the `request virtual-chassis renumber` command.

## RELATED DOCUMENTATION

[Example: Configuring an EX3300 Virtual Chassis with a Primary and Backup | 91](#)

[Configuring Primary Role of a Virtual Chassis](#)

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

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

### IN THIS SECTION

- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis with a Preprovisioned Configuration File | 246](#)
- [Configuring an EX4200, an EX4500, or EX4550 Virtual Chassis with a Nonprovisioned Configuration File | 248](#)

This topic does not apply to a mixed Virtual Chassis configuration. For information on configuring a mixed Virtual Chassis, see ["Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)"](#) on page 255.

You can interconnect EX4200, EX4500, and EX4550 switches using the dedicated Virtual Chassis ports (VCPs) on the rear panel of EX4200 switches and on the Virtual Chassis module in EX4500 and EX4550 switches. You do not have to configure the interfaces for the dedicated VCPs.

If you want to interconnect member switches that are located in different racks or wiring closets, you interconnect them using optical ports configured as VCPs. See *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port* or ["Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\)"](#) on page 293.

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

An EX4200, EX4500, or EX4550 Virtual Chassis can be configured with either:

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

This topic includes:

## Configuring an EX4200, EX4500, or EX4550 Virtual Chassis with a Preprovisioned Configuration File

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

To configure a Virtual Chassis using a preprovisioned configuration:

1. Make a list of the serial numbers of all the switches to be connected in the Virtual Chassis configuration.
2. Note the desired role (**routing-engine** or **line-card**) you want for each switch. If you configure the member with a **routing-engine** role, it is eligible to function as a primary or backup. If you configure the member with a **line-card** role, it is not eligible to become a primary or backup.
3. Interconnect the member switches using the dedicated VCPs. See *Connecting a Virtual Chassis Cable to an EX4200 Switch* or [Connecting a Virtual Chassis Cable to an EX4500 Switch](#).

**NOTE:** For management purposes, we recommend arranging the switches in member ID sequence, either from top to bottom or from bottom to top (0–9).

4. Power on only the switch that you plan to use as the primary switch. Do not power on the other switches at this time.
5. Run the EZSetup program on the primary switch, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.

**NOTE:** The properties that you specify for the primary switch apply to the entire Virtual Chassis configuration, including all the members listed in the preprovisioned configuration file.

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

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

7. Specify the preprovisioned configuration mode:

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

8. From the primary switch, specify all the members that you want to include in the Virtual Chassis configuration, listing each switch's serial number with the desired member ID and the desired role:

```
[edit virtual-chassis]
user@switch# set member 0 serial-number abc123 role routing-engine
user@switch# set member 1 serial-number def456 role line-card
user@switch# set member 2 serial-number ghi789 role line-card
user@switch# set member 3 serial-number jkl012 role line-card
user@switch# set member 4 serial-number mno345 role line-card
user@switch# set member 5 serial-number pqr678 role routing-engine
user@switch# set member 6 serial-number stu901 role line-card
user@switch# set member 7 serial-number vwx234 role line-card
user@switch# set member 8 serial-number yza567 role line-card
user@switch# set member 9 serial-number bcd890 role line-card
```

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

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

10. Power on the member switches.
11. (EX4500 switches only) Verify the PIC mode setting:

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

If the PIC mode setting is not set to **virtual-chassis**, set the PIC mode to **virtual-chassis**:

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

**NOTE:** This step is required if you are using the dedicated VCPs on the Virtual Chassis module to connect the Virtual Chassis.

The PIC mode setting has no impact on uplink ports that are configured as VCPs.

**NOTE:** You cannot modify the primary-role priority when you are using a preprovisioned configuration. The primary-role priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file. The two Routing Engines are

assigned the same primary-role priority value. However, the member that was powered on first has higher prioritization according to the primary-role election algorithm. See *Understanding How the Primary in a Virtual Chassis Is Elected*.

## Configuring an EX4200, an EX4500, or EX4550 Virtual Chassis with a Nonprovisioned Configuration File

To configure the Virtual Chassis using a nonprovisioned configuration:

1. Interconnect the member switches using the dedicated VCPs. See *Connecting a Virtual Chassis Cable to an EX4200 Switch* or [Connecting a Virtual Chassis Cable to an EX4500 Switch](#).

**NOTE:** For management purposes, we recommend arranging the switches in member ID sequence, either from top to bottom or from bottom to top (0–9).

2. Power on only the switch that you plan to use as the primary switch (SWA-0). Do not power on the other switches at this time.
3. Run the EZSetup program on SWA-0, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details.

**NOTE:** The properties that you specify for SWA-0 apply to the entire Virtual Chassis configuration, including all the members interconnected through VCPs.

4. (Optional) Configure SWA-0 with the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis configuration, if desired:

```
[edit]
user@SWA-0# set interfaces vme unit 0 family inet address /ip-address/mask/
```

5. Configure primary-role priority for the primary, backup, and other members, if desired:

```
[edit virtual-chassis]
user@SWA-0# set member 0 mastership-priority 255
user@SWA-0# set member 5 mastership-priority 255
```

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

```
[edit virtual-chassis]
user@SWA-0# set no-split-detection
```

7. Power on the member switches in sequential order, one by one.
8. (EX4500 switches only) Enter the `show chassis pic-mode operational` command to verify the current PIC mode setting. If the PIC mode is currently set to **intraconnect**, enter the **request chassis pic-mode virtual-chassis** operational mode command to set the PIC mode to **virtual-chassis**. Reboot the switch to complete the procedure.

**NOTE:** This step is only required if you are using the dedicated VCP ports on the Virtual Chassis module.

**NOTE:** If you do not edit the Virtual Chassis configuration file, a nonprovisioned configuration is generated by default. The primary-role priority value for each member switch is **128**. The primary role is selected by default. You can change the role that is performed by the members by modifying the primary-role priority. See *Configuring Primary Role of a Virtual Chassis*. We recommend that you specify the same primary-role priority value for the desired primary and backup members. In this example, the highest possible primary-role priority has been assigned to two members. However, the member that was powered on first has higher prioritization according to the primary-role election algorithm. See *Understanding How the Primary in a Virtual Chassis Is Elected*. The other members use the default primary-role priority in this example, which configures them to function in the role of linecard.

**NOTE:** If you want to change the member ID that the primary has assigned to a member switch, use the `request virtual-chassis renumber` command.

## RELATED DOCUMENTATION

[Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis | 183](#)

[Configuring Primary Role of a Virtual Chassis](#)

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

## Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure)

### IN THIS SECTION

- [Configuring an EX2200, EX2200-C, EX3300, EX4200, EX4300, EX4500, EX4550, or EX4600 Virtual Chassis \(J-Web Procedure\) | 250](#)
- [Enabling Virtual Chassis Mode on an EX8200 Switch \(J-Web Procedure\) | 252](#)
- [Configuring an EX8200 Virtual Chassis \(J-Web Procedure\) | 252](#)

### Configuring an EX2200, EX2200-C, EX3300, EX4200, EX4300, EX4500, EX4550, or EX4600 Virtual Chassis (J-Web Procedure)

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

To take advantage of the scalability features of EX2200, EX2200-C, EX3300, EX4200, EX4300, EX4500, EX4550, and EX4600 switches, you can configure a Virtual Chassis. EX2200 and EX2200-C Virtual Chassis can include up to four member switches. EX3300 Virtual Chassis include up to six member switches. For Junos OS Release 14.1X53-D10 and later, EX3300 Virtual Chassis include up to 10 member switches. EX4200, EX4300, EX4500, EX4550, and EX4600 Virtual Chassis can include up to 10 member switches. You can interconnect the member switches by using dedicated Virtual Chassis ports (VCPs). You do not have to configure the interface for the dedicated VCPs. If you want to interconnect member switches that are located in different racks or wiring closets, interconnect them using 10-gigabit ports (SFP+ uplink module ports or SFP+ network ports for EX4500) configured as VCPs. EX4300, EX4550, and EX4600 switches support 10-gigabit (SFP+ expansion module ports) and 40-gigabit (QSFP + uplink module ports or fixed ports) Ethernet interface. See *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*.

**NOTE:** You can form a Virtual Chassis using EX4600 and EX4300 switches. When in a mixed Virtual Chassis consisting of EX4600 switches and EX4300 switches, the EX4600 switches can be the primary, backup, or in the linecard role, while the EX4300 switches must be in the linecard role.

To configure a Virtual Chassis by using the J-Web interface:

1. Select **Configure > Virtual Chassis**.

**NOTE:** After you make changes to the configuration on this page, you must commit the changes for them to take effect. To commit all changes to the active configuration, select **Commit Options > Commit**. See Using the Commit Options to Commit Configuration Changes for details about all commit options.

2. The properties that you can configure are displayed.
- The first section of the Virtual Chassis Configuration page displays the Virtual Chassis member configuration. The display includes a list of member switches, their member IDs, and the primary-role priority.
- The second section displays the operational status of the Virtual Chassis configuration, member details, and the dedicated and the configured VCPs.
3. Enter information into the page as described in [Table 24 on page 251](#).
4. Click one of the following options:
- **Add**—To add a member's configuration to the Virtual Chassis configuration, click **Add**.
  - **Edit**—To modify an existing member's configuration, click **Edit**.
  - **Delete**—To delete the configuration of a member, click **Delete**.
5. To configure uplink ports on EX2200, EX2200-C, and EX3300 switches and uplink module ports on EX4200, EX4300, EX4500, and EX4600 switches as VCPs , select the member in the Virtual Chassis members list and select **Action > Set Uplink Port as VCP**. Select the port from the list.
- On EX4550 switches, to configure a VCP, select the member in the Virtual Chassis members list and select **Action > Set Port as VCP**. Select the port from the list.
6. To remove the VCP configuration from the uplink ports on EX2200, EX2200-C, and EX3300 switches and uplink module ports on EX4200, EX4300, EX4500, and EX4600 switches , select the member in the Virtual Chassis members list and select **Action > Delete Uplink Port as VCP**.
- On EX4550 switches, to remove the VCP configuration from the port of a member, select the member in the Virtual Chassis members list and select **Action > Delete Port as VCP**.

Table 24: Virtual Chassis Configuration Fields

Field	Function	Your Action
Member Details		



Table 24: Virtual Chassis Configuration Fields *(Continued)*

Field	Function	Your Action
Member ID	Specifies the identifier for the member switch.	Select an identifier (from <b>0</b> through <b>9</b> ) from the list.  <b>NOTE:</b> For EX2200 Virtual Chassis, you can select the member ID (from <b>0</b> through <b>3</b> ) from the list.
Priority	Specifies the primary-role priority to be assigned to the member.	Select a number from <b>1</b> through <b>255</b> , (255 being the highest priority and <b>128</b> , the default).
Disable Management VLAN	If you want to reserve an individual member's management Ethernet port, you can remove that port from being part of the virtual management Ethernet (VME) interface.	Click to disable the management VLAN on the port.
Refresh	Refreshes the operational status of Virtual Chassis members.	Click to refresh the operational status.

## Enabling Virtual Chassis Mode on an EX8200 Switch (J-Web Procedure)

Using the J-Web interface, you can enable Virtual Chassis mode on an EX8200 switch. To enable Virtual Chassis mode:

1. Select **Configure > Virtual Chassis**. The Virtual Chassis page displays the serial number of the member switch. You need the serial number while preprovisioning the Virtual Chassis.
2. Click **Enable Virtual Chassis**.

## Configuring an EX8200 Virtual Chassis (J-Web Procedure)

### IN THIS SECTION

- [Preprovision the Virtual Chassis | 253](#)
- [Configure Virtual Chassis Members | 253](#)

Using the J-Web interface, 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 the management ports 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.

The EX8200 Virtual Chassis wizard helps to preprovision Virtual Chassis members and to configure Virtual Chassis ports.

To configure an EX8200 Virtual Chassis, select **Configure > Virtual Chassis**. The Virtual Chassis wizard is displayed. You can:

### Preprovision the Virtual Chassis

If the Virtual Chassis has not been preprovisioned:

1. Select **Preprovision Virtual Chassis**. The prerequisites page is displayed.
2. Ensure that the following prerequisites are met:
  - The same version of Junos OS is running on all XRE200 Routing Engines and switches.
  - Virtual Chassis mode has been enabled on each EX8200 switch.
  - The XRE200 Routing Engines and switches have been cabled and connected.

Click **Next**. The Configure Virtual Chassis Members screen is displayed.

### Configure Virtual Chassis Members

When you click **Configure Virtual Chassis Members** in the wizard, you can add a Virtual Chassis member, modify an existing Virtual Chassis member configuration, or delete the Virtual Chassis configuration for an existing member.

To add a Virtual Chassis member, click **Add**. For each member, specify:

- **Member ID**—The identifier for the member switch or the XRE200 External Routing Engine.
- **Serial Number**—The serial number of the member switch or the XRE200 External Routing Engine.
- **Management VLAN**—Click to enable or disable the management VLAN on the port.

- **Location**—A description of the location of the EX8200 member switch or external Routing Engine.
- **Role**—The role to be performed by each EX8200 member switch or each XRE200 External Routing Engine. EX8200 switches must be in the linecard role and XRE200 External Routing Engines must be in the Routing Engine role.

To edit an existing member configuration, click **Edit**.

**NOTE:** If you are editing an existing member configuration, you can modify only the Management VLAN option and the location of the Virtual Chassis member.

Click **Remove** to delete the Virtual Chassis configuration for an existing member.

**Configure Virtual Chassis Ports**

To configure Virtual Chassis ports that are needed between the switches for data traffic, select the **Configure Virtual Chassis Ports** option in the wizard.

**NOTE:** All XRE200 ports, the vcp-0/0 port, and any port on a Virtual Chassis Control Interface (VCCI) module are automatically VCPs. You need not configure these ports as VCPs.

1. Select the member from the list. The switch model, serial number, available ports, and configured ports are displayed. Select **All** to view details of available ports and configured ports of all the members.

**NOTE:** Only members with active EX8200-8XS line cards are listed. The J-Web interface does not support EX8200-2X4F40TE/PE line cards.

2. To convert network ports to Virtual Chassis ports or to convert Virtual Chassis ports to network ports, click the appropriate arrow.
3. Click **Next**. The Summary page displays the configuration changes. You can edit the configuration.

**Release History Table**

Release	Description
14.1X53-D10	For Junos OS Release 14.1X53-D10 and later, EX3300 Virtual Chassis include up to 10 member switches.

## RELATED DOCUMENTATION

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

[Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) | 293](#)

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

[Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets | 148](#)

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

*Virtual Chassis Cabling Configuration Examples for EX4200 Switches*

*Understanding EX Series Virtual Chassis*

*Configuring an EX8200 Virtual Chassis (CLI Procedure)*

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

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

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

## Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)

This topic explains how to configure a mixed Virtual Chassis composed of any combination of EX4200, EX4500, and EX4550 switches. For information on configuring an EX4200 Virtual Chassis, EX4500 Virtual Chassis, or EX4550 Virtual Chassis, see "[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)](#)" on page 245.

You can configure the following mixed Virtual Chassis:

- Mixed EX4200 and EX4500 Virtual Chassis
- Mixed EX4200 and EX4550 Virtual Chassis
- Mixed EX4200, EX4500, and EX4550 Virtual Chassis
- Mixed EX4500 and EX4550 Virtual Chassis

You can configure up to ten member switches in all mixed Virtual Chassis composed of EX4200, EX4500, and EX4550 switches. You can interconnect and mix any model of EX4200, EX4500, or EX4550 switch into the same mixed Virtual Chassis, and any EX4200, EX4500, or EX4550 switch can

assume the primary, backup, or linecard role. You can interconnect the member switches using the dedicated Virtual Chassis ports (VCPs) or by configuring optical ports as VCPs.

This procedure uses a preprovisioned Virtual Chassis configuration. A preprovisioned configuration allows you to deterministically control the member ID and role assigned to a member switch by tying it to its serial number.

Nonprovisioned configuration is also supported for a mixed Virtual Chassis configuration. In a nonprovisioned configuration, the primary sequentially assigns a member ID to other member switches and the other member switches roles are determined by the primary-role priority value and other factors in the primary-role election algorithm. See *Understanding How the Primary in a Virtual Chassis Is Elected*.

Ensure that all switches are running the same version of Junos OS. See [Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine \(CLI Procedure\)](#).

To configure a mixed Virtual Chassis:

1. Make a list of the serial numbers of all the switches to be connected in the Virtual Chassis. You can get the serial numbers in the `show chassis hardware` output or by following the instructions in *Locating the Serial Number on an EX4200 Switch or Component*, [Locating the Serial Number on an EX4500 Switch or Component](#), or [Locating the Serial Number on an EX4550 Switch or Component](#).
2. Decide which switches you want to act in the primary and backup roles. See *Understanding Virtual Chassis Components*.
3. Power and log in to the switch that you want to function in the primary role. If the switch has not previously been configured, configure it. See *Connecting and Configuring an EX Series Switch (CLI Procedure)*.
4. Power and log in to the switch that you want to function in the backup role. If the switch has not previously been configured, configure it. See *Connecting and Configuring an EX Series Switch (CLI Procedure)*.
5. Power on all other member switches.
6. (EX4500 switches only) Verify the PIC mode setting:

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

7. (EX4500 switches only) If the PIC mode was not set to Virtual Chassis mode, set the PIC mode to Virtual Chassis mode:

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

**NOTE:** This step is required if you are using the dedicated VCPs on the Virtual Chassis module to connect the Virtual Chassis.

The PIC mode setting has no impact on optical ports that are configured as VCPs.

8. (Mixed Virtual Chassis with EX4200 switches only) Set the Virtual Chassis mode to mixed on all member switches if your mixed Virtual Chassis includes EX4200 switches:

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

9. If the PIC mode or the mixed mode was changed in step 7 or step 8, reboot each member switch:

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

You can skip this step and proceed to step 11 if you did not change the PIC mode or the mixed mode.

10. After you have rebooted the switches, log into the switch that you powered on first. This switch is the primary switch.
11. Run the EZSetup program on the primary switch, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)*.

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

12. (Optional) On the primary switch, configure the virtual management Ethernet (VME) interface for out-of-band management of the Virtual Chassis, if desired:

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

13. On the primary switch, specify the preprovisioned configuration mode:

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

14. On the primary switch, specify all members for the Virtual Chassis configuration, listing each switch's serial number with the desired member ID and the desired role.

**NOTE:** The **routing-engine** role can be assigned to any member switch, unless otherwise noted.

The **routing-engine** role can be assigned to any member switch in a mixed Virtual Chassis starting in Junos OS Release 11.4.

Ensure that you assign the **routing-engine** role to two members that are the same type of switch if you are running Junos OS Release 11.2 or 11.3..

Ensure that you assign only the **routing-engine** role to EX4500 member switches if you are running Junos OS Release 11.1.

```
[edit virtual-chassis]
user@switch# set member 0 serial-number serial-number role routing-engine
user@switch# set member 1 serial-number serial-number role routing-engine
user@switch# set member 2 serial-number serial-number role line-card
user@switch# set member 3 serial-number serial-number role line-card
user@switch# set member 4 serial-number serial-number role line-card
user@switch# set member 5 serial-number serial-number role line-card
user@switch# set member 6 serial-number serial-number role line-card
user@switch# set member 7 serial-number serial-number role line-card
user@switch# set member 8 serial-number serial-number role line-card
user@switch# set member 9 serial-number serial-number role line-card
```

15. Interconnect the member switches by using either the dedicated VCPs on the member switches (see *Connecting a Virtual Chassis Cable to an EX4200 Switch* or [Connecting a Virtual Chassis Cable to an EX4500 Switch](#)) or by connecting them through the optical ports that you have configured as VCPs (see *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port* or ["Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\)" on page 293](#)).

**NOTE:** You cannot modify the primary-role priority when you are using a preprovisioned configuration. The primary-role priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file.

For information on adding a switch to an existing Virtual Chassis or configuring a Virtual Chassis port (VCP) to connect member switches over long distances, see *Related Documentation*.

## RELATED DOCUMENTATION

*Configuring Primary Role of a Virtual Chassis*

*Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)*

[Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) | 269](#)

[Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 271](#)

[Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 273](#)

*Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*

[Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) | 293](#)

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

## Installing Software on a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Switches (CLI Procedure)

You can use this procedure to upgrade or downgrade Junos OS for all member switches in an any operational mixed Virtual Chassis composed of any combination of EX4200, EX4500, or EX4550 switches.

To upgrade Junos OS for an operational mixed Virtual Chassis:

1. Download the same version of Junos OS software for each switch in the mixed Virtual Chassis.

**NOTE:** You must download Junos OS Release 11.4 or later if you have three or more EX4500 switches in your mixed Virtual Chassis.

**NOTE:** EX4500 and EX4550 switches run the same Junos OS. You only need to download one Junos OS image if you are running a mixed EX4500 and EX4550 Virtual Chassis.

2. (Optional) Back up the current software configuration to a second storage option. See the [Junos OS Installation and Upgrade Guide](#) for instructions on performing this task.
3. (Optional) Copy the software packages to the member switch acting in the primary role. We recommend that you use FTP to copy the file to the `/var/tmp` directory.

This step is optional because Junos OS can also be upgraded when the software image is stored at a remote location. These instructions describe the software upgrade process for both scenarios.



#### 4. Install the new packages on the Virtual Chassis:

```
user@switch> request system software add set [package (package)]
```

**NOTE:** You enter the `request system software add set [package package]` command once on the Virtual Chassis to download the software package onto all member switches in the mixed Virtual Chassis. If your mixed Virtual Chassis includes at least one EX4200 members switch, ensure that the *package* path points to one EX4200 Junos OS image and, one EX4500 Junos OS image.

You only need to download one EX4500 switch Junos OS image to upgrade a mixed Virtual Chassis because a Junos OS image for an EX4500 switch can be used to upgrade EX4500 and EX4550 switches.

Replace *package* with one of the following paths:

- For a software package in a local directory on the switch—`/var/tmp/package.tgz`.
- For a software package on a remote server:
  - `ftp://hostname/pathname/package.tgz`
  - `http://hostname/pathname/package.tgz`

where *package*.tgz is, for example, `jinstall-ex-4200-11.1R1.8-domestic-signed.tgz`.

**NOTE:** To terminate the installation, do not reboot your Virtual Chassis; instead, finish the installation and then issue the `request system software delete package.tgz` command, where *package*.tgz is, for example, `jinstall-ex-4500-11.1R1.8-domestic-signed.tgz`. This is your last chance to stop the installation.

#### 5. Reboot the Virtual Chassis to start the new software:

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

#### 6. After the reboot has completed, log in and verify that the new version of the software is properly installed for all member switches in the Virtual Chassis:

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

## RELATED DOCUMENTATION

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

*Understanding Virtual Chassis Components*

## Adding a New Switch to an Existing EX3300 Virtual Chassis (CLI Procedure)

You can use this procedure to add an EX3300 switch to an EX3300 Virtual Chassis.

Before you begin, be sure you have:

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

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

- (Optional) Configured Ethernet interfaces on different member switches into the same LAG. See ["Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch" on page 191](#)

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

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

1. If the new member switch has been previously configured, revert that switch's configuration to the factory defaults. See *Reverting to the Default Factory Configuration for the EX Series Switch*.

2. Interconnect the unpowered new switch to one member of the existing Virtual Chassis configuration using either uplink port 2 or 3. Uplink ports 2 and 3 are configured as VCPs by default. See [Virtual Chassis Cabling Configuration Examples for EX3300 Switches](#).

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

**NOTE:** Because EX3300 switches use uplink ports configured as VCPs to form a Virtual Chassis, this step is identical for short and long distance (up to 6.2 miles) VCP connections.

3. Power on the new switch.
4. Confirm that the new member switch is now included within the Virtual Chassis configuration by checking the front-panel LCD for the member ID. It should display a member ID that is greater than 0 (1 through 5), because there is already at least one member of the Virtual Chassis configuration.

**NOTE:** If you are using a preprovisioned configuration, the member ID is automatically assigned to the member's serial number in the configuration file.

5. Cable the other default VCP - either uplink port 2 or 3 - on the new member switch to the Virtual Chassis.



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

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

## RELATED DOCUMENTATION

| [Example: Expanding an EX3300 Virtual Chassis](#) | 112

## Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)

### IN THIS SECTION

- [Adding a New Switch to an Existing Virtual Chassis Within the Same Wiring Closet | 263](#)
- [Adding a New Switch from a Different Wiring Closet to an Existing Virtual Chassis | 264](#)
- [Adding a New Switch to an Existing Preprovisioned Virtual Chassis Using Autoprovisioning and Automatic VCP Conversion | 266](#)

This topic explains how to add an EX4200 switch to an existing EX4200 Virtual Chassis. For information about adding an EX4200 switch to an EX4500 Virtual Chassis or a mixed Virtual Chassis, see [Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\)](#).

To add an EX4200 switch to an existing EX4200 Virtual Chassis, use the procedure that matches what you need to accomplish:

### Adding a New Switch to an Existing Virtual Chassis Within the Same Wiring Closet

This procedure can be used to add an EX4200 switch to an EX4200 Virtual Chassis.

Before you begin, be sure you have:

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

**NOTE:** After you have created a preprovisioned Virtual Chassis configuration, you can use the autoprovisioning feature to add member switches to that configuration.

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

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

To add a new member switch to an existing Virtual Chassis configuration within the same wiring closet:

1. If the new member switch has been previously configured, revert that switch's configuration to the factory defaults. See *Reverting to the Default Factory Configuration for the EX Series Switch*.
2. Interconnect the unpowered new switch to one member of the existing Virtual Chassis configuration using the dedicated Virtual Chassis ports (VCPs). Connect only one VCP on the unpowered new switch to a VCP on a member switch in the existing Virtual Chassis at this point of the procedure.
3. Power on the new switch.
4. Confirm that the new member switch is now included within the Virtual Chassis configuration by checking the front-panel LCD for the member ID. It should display a member ID that is greater than 0 (1 through 9), because there is already at least one member of the Virtual Chassis configuration.

**NOTE:** If you are using a preprovisioned configuration, the member ID is automatically assigned to the member's serial number in the configuration file.

5. Cable the other dedicated VCP on the new member switch to the Virtual Chassis.



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

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

## Adding a New Switch from a Different Wiring Closet to an Existing Virtual Chassis

This procedure can be used to add an EX4200 switch to an EX4200 Virtual Chassis from a different wiring closet to an existing Virtual Chassis.

To add a new switch from a different wiring closet to an existing Virtual Chassis configuration, you must use a long cable to connect the members switches across wiring closets. You can use any SFP, SFP+, or XFP port and a fiber-optic cable for this purpose.

Before you begin, be sure you have:

- Installed the uplink modules needed for the Virtual Chassis configuration. See *Installing an Uplink Module in an EX4200 Switch*.
- Mounted the new switch in a rack.
- If the new member switch has been previously configured, reverted its configuration to the factory defaults. See *Reverting to the Default Factory Configuration for the EX Series Switch*.
- If you are expanding a preprovisioned configuration, made a note of the serial number (the number is on the back of the switch). You will need to edit the Virtual Chassis configuration to include the serial number of the new member switch.
- If you are expanding a preprovisioned configuration, edited the existing Virtual Chassis configuration to include the serial number of the new member switch. You can specify the role of the new member switch when you add its serial number in the Virtual Chassis configuration file. The parameters specified in the primary Virtual Chassis configuration file are applied to the new switch after it has been interconnected with its uplink VCP to an existing member switch.

**NOTE:** After you have created a preprovisioned Virtual Chassis configuration, you can use the autoprovisioning feature to add member switches to that configuration.

To add a new member switch that is going to be interconnected with the existing Virtual Chassis configuration across wiring closets:

1. Power on the new switch.
2. Connect a laptop or terminal to the console port of the switch, or use EZSetup on the LCD Panel of the standalone switch to specify temporary identification parameters. (When you interconnect the new member switch with the existing Virtual Chassis configuration, the primary will overwrite and disable any specified parameters that conflict with the Virtual Chassis parameters or assigned member configuration.)
3. Use the CLI or the J-Web interface to set one uplink module port as a VCP:

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

**NOTE:** If you are using a nonprovisioned configuration, you might configure the new member switch with a primary-role priority value that is less than that of the existing member

switches. Doing so ensures that the new member switch will function in a linecard role when it is included within the Virtual Chassis configuration.

4. Power off the new switch.
5. Interconnect the new member switch to one existing member switch in the Virtual Chassis configuration using one of the uplink module ports that you have configured as a VCP. Connect only one VCP on the unpowered new switch to a VCP on a member switch in the existing Virtual Chassis at this point of the procedure.
6. Power on the new member switch.
7. Confirm that the new member switch is now included within the Virtual Chassis configuration by checking the front-panel LCD for the member ID. It should display a member ID that is greater than 0 (1 through 9), because there is already at least one member of the Virtual Chassis configuration.

**NOTE:** If you are using a preprovisioned configuration, the member ID is automatically assigned to the member's serial number in the configuration file.

8. Cable another user-configured VCP on the new member switch to the Virtual Chassis, if desired.



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

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

## Adding a New Switch to an Existing Preprovisioned Virtual Chassis Using Autoprovisioning and Automatic VCP Conversion

This procedure can be used to add an EX4200 switch to an existing EX4200 Virtual Chassis using autoprovisioning, where after the new switch is provisioned and cabled into the Virtual Chassis, the interconnecting links automatically convert into VCP links.

Before you begin, be sure you have:

- Installed the uplink modules needed for the Virtual Chassis configuration.
- Mounted the new switch in a rack.

- Ensured that the preprovisioned Virtual Chassis configuration has an active primary switch. For more information, see [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File](#).
- On the primary switch, configured the Link Level Discovery Protocol (LLDP) on the uplink module ports that will be used as VCPs. LLDP is configured by default but might have been disabled. Both sides of the new links exchange LLDP messages to accomplish automatic VCP conversion. To configure LLDP, see [Configuring LLDP \(CLI Procedure\)](#).
- Ensured that the new member switch has the factory-default configuration. If the new member switch has been previously configured, revert its configuration to the factory defaults. See *Reverting to the Default Factory Configuration for the EX Series Switch*.
- Made a note of the serial number (the number is on the back of the switch). You will need to edit the Virtual Chassis configuration to include the serial number of the new member switch.
- Edited the existing Virtual Chassis preprovisioned configuration to include the serial number of the new member switch. The parameters specified in the primary Virtual Chassis configuration file are applied to the new member switch after it has been interconnected through its uplink VCP to an existing member switch.
- Prepared an existing member switch to interconnect with the new switch through an uplink module port by ensuring that port is not set as a VCP. One of the conditions for automatic VCP conversion to be invoked is that the ports on both sides of the new link must not already be configured as VCPs.



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

- Ensured that the operational modes of the uplink modules on the existing member switch and the new member switch match.
- Confirmed that the new member switch is powered off.

If the preceding conditions are not met, autoprovisioning will not work and you will need to manually configure uplink module ports on the switch to be added to the configuration to be VCPs. For more information, see [Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port](#).



To add a switch to an existing preprovisioned Virtual Chassis configuration using the autoprovisioning feature:

1. Interconnect the unpowered new switch to one member of the existing Virtual Chassis configuration. Only connect one VCP on the unpowered new switch to a VCP on a member switch in the existing Virtual Chassis at this point of the procedure.
2. Power on the new member switch.
3. Confirm that the new member switch is now included in the Virtual Chassis configuration by checking the front-panel LCD for the member ID. It should display a member ID in the range from 0 through 9. The member ID is automatically assigned to the new member switch's serial number in the configuration file.
4. Cable the other VCP on the new member switch to the Virtual Chassis.



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

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

## RELATED DOCUMENTATION

---

[Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet](#)

---

[Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration](#)

---

[Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets](#)

---

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

---

[Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches](#)

---

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

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

This topic explains how to add an EX4200 switch to an already configured and operational EX4500 Virtual Chassis or to a mixed EX4200 and EX4500 Virtual Chassis that was configured using a preprovisioned configuration.

To add an EX4200 switch to an EX4500 Virtual Chassis or a mixed EX4200 and EX4500 Virtual Chassis that was configured using a preprovisioned configuration:

1. Power on the EX4200 switch that will be added to the Virtual Chassis.
2. Set the Virtual Chassis mode to mixed:

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

3. Reboot the EX4200 switch.

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

4. Log in to the EX4500 Virtual Chassis or the mixed EX4200 and EX4500 Virtual Chassis.
5. (EX4500 Virtual Chassis only) Set the Virtual Chassis mode to mixed for all Virtual Chassis member switches:

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

**NOTE:** Each member switch will already have been configured in mixed mode in a mixed EX4200 and EX4500 Virtual Chassis.

6. (EX4500 Virtual Chassis only) Reboot all member switches in the EX4500 Virtual Chassis:

```
user@switch> request system reboot all-members
```

7. Add the EX4200 switch to the preprovisioned configuration:

```
[edit virtual-chassis]
user@SWA-0# set member 5 serial-number serial-number role role
```

8. If you are replacing an EX4500 switch in the **routing-engine** role with an EX4200 switch, change the role of the EX4500 switch to **line-card**:

```
[edit virtual-chassis]
user@switch# replace member 0 serial-number serial-number role routing-engine with member 0
serial-number serial-number role line-card
user@switch# replace member 1 serial-number serial-number role routing-engine with member 1
serial-number serial-number role line-card
```

9. Commit the configuration:

```
user@switch# commit synchronize
```

10. Cable one dedicated or user-configured Virtual Chassis port (VCP) on the new member switch to the existing Virtual Chassis. See *Connecting a Virtual Chassis Cable to an EX4200 Switch* or *Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*.
11. Wait for the new switch to become operational in the Virtual Chassis. Monitor the `show virtual-chassis` command output to confirm the new switch is recognized by the Virtual Chassis and is in the **Prsnt** state.



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

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

12. Cable the other dedicated or user-configured VCP on the new member switch to the Virtual Chassis.

## RELATED DOCUMENTATION

*Configuring Primary Role of a Virtual Chassis*

*Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port*

[Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) | 293](#)

## Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis (CLI Procedure)

An EX4500 switch can be added to an EX4200 Virtual Chassis to create a mixed EX4200 and EX4500 Virtual Chassis. This topic explains how to add an EX4500 switch to an already configured and operational EX4200 Virtual Chassis.

To add an EX4500 switch to an EX4200 Virtual Chassis that was configured using a preprovisioned configuration:

1. Power on the EX4500 switch with the installed Virtual Chassis module.
2. Verify the PIC mode setting:

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

- 3.

**NOTE:** Step 3 is required only if you are using the dedicated VCPs on the Virtual Chassis module of the EX4500 switch to connect the Virtual Chassis.

If the PIC mode was not set to Virtual Chassis mode, set the PIC mode to Virtual Chassis mode:

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

4. Set the Virtual Chassis mode to mixed:

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

5. Reboot the EX4500 switch.

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

6. Log in to the EX4200 Virtual Chassis.
7. Set the Virtual Chassis mode to mixed for all member switches:

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

8. Reboot the Virtual Chassis.

```
user@switch> request system reboot all-members
```

9. Log in to the EX4200 Virtual Chassis after the reboot is complete.
10. On the primary switch, add the EX4500 switch to the preprovisioned configuration:

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

11. If you are using the EX4500 switch in the primary role in place of an EX4200 switch, change the EX4200 switch in the **routing-engine** role to the **line-card** role:

```
[edit virtual-chassis]
user@switch# replace member 0 serial-number serial-number role routing-engine with member 0
serial-number serial-number role line-card
user@switch# replace member 1 serial-number serial-number role routing-engine with member 1
serial-number serial-number role line-card
```

12. Commit the configuration:

```
user@switch# commit synchronize
```

13. Interconnect the EX4500 switch to one member of the existing Virtual Chassis configuration. See *Connecting a Virtual Chassis Cable to an EX4200 Switch* and [Connecting a Virtual Chassis Cable to an EX4500 Switch](#).

Connect only one VCP on the EX4500 switch to a VCP on a member switch in the existing Virtual Chassis at this point of the procedure.

14. Confirm that the new member switch is now included in the Virtual Chassis configuration by checking the front-panel LCD for the member ID. It should display a member ID in the range from 0 through 9. The member ID is automatically assigned to the new member switch's serial number in the configuration file.
15. Cable the other VCP on the Virtual Chassis of the EX4500 switch to the Virtual Chassis.



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

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

## RELATED DOCUMENTATION

[Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 273](#)

[Configuring Primary Role of a Virtual Chassis](#)

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

[Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port](#)

## Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis (CLI Procedure)

An EX4500 switch can be added to an EX4200 Virtual Chassis to create a mixed EX4200 and EX4500 Virtual Chassis.

This topic explains how to add an EX4500 switch to an already configured and operational EX4200 Virtual Chassis that was configured using a nonprovisioned configuration.

Before you begin adding EX4500 switches to the Virtual Chassis:

- Ensure that a Virtual Chassis module is installed in the EX4500 switch.

To add an EX4500 switch to an EX4200 Virtual Chassis that was configured using a nonprovisioned configuration:

1. Power on the EX4500 switch.
2. Verify the PIC mode setting:

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

**NOTE:** This step is required only if you are using the dedicated VCPs on the Virtual Chassis module of the EX4500 switch to connect the Virtual Chassis.

3. If the PIC mode was not set to Virtual Chassis mode, set the PIC mode to Virtual Chassis mode:

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

4. Set the Virtual Chassis mode to mixed:

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

5. Reboot the EX4500 switch.

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

6. Log in to the EX4200 Virtual Chassis.
7. Set the Virtual Chassis mode to mixed for all member switches:

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

8. Reboot all member switches in the Virtual Chassis.

```
user@switch> request system reboot all-members
```

9. After all members have rebooted, interconnect one dedicated VCP on the EX4500 switch with one dedicated VCP of a member EX4200 switch in the EX4200 Virtual Chassis. . See *Connecting a Virtual Chassis Cable to an EX4200 Switch* and [Connecting a Virtual Chassis Cable to an EX4500 Switch](#).

**NOTE:** You can also connect an EX4500 switch into the Virtual Chassis by configuring an uplink port as a VCP. See ["Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\)"](#) on page 293.

10. Log in to the Virtual Chassis.
11. Wait for the new switch to become operational in the Virtual Chassis. Monitor the `show virtual-chassis` command output to confirm the new switch is recognized by the Virtual Chassis and is in the **Prsnt** state.



**CAUTION:** If you immediately cable both VCPs on the new switch into the existing Virtual Chassis at the same time, a member switch that was already part of the

Virtual Chassis might become nonoperational for several seconds. Network traffic to this switch is dropped during the downtime.

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

12. Cable the other dedicated or user-configured VCP on the new member switch to the Virtual Chassis.
13. (Optional) Set the primary-role priority of the EX4500 switch. The switches with the highest primary role priorities assume the primary role and backup roles.

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

14. (Optional) From the primary switch, set the primary-role priority of the other member switches:

```
[edit virtual-chassis]
user@switch# replace member member-id mastership-priority mastership-priority with member
member-id mastership-priority mastership-priority
```

## RELATED DOCUMENTATION

[Example: Adding an EX4500 Switch to a Nonprovisioned Virtual Chassis | 134](#)

[Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) | 269](#)

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

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

### IN THIS SECTION

- [Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch | 277](#)



- [Remove, Repair, and Reinstall the Same Switch | 278](#)
- [Remove a Member Switch, Replace It with a Different Switch, and Reapply the Old Configuration | 279](#)
- [Replace a Member Switch With a Different Type of Switch That Changes the Virtual Chassis to Mixed Mode | 282](#)

You can remove or replace a member switch in a Virtual Chassis without disrupting network service on the other member switches.

If you remove a member switch, you can free up the member ID so it is available to be assigned to a new member switch later.

When you add a new member switch, the Virtual Chassis assigns the next available member ID to it. The Virtual Chassis retains the existing configuration items specific to particular member IDs. The Virtual Chassis applies those items to a replacement member switch that has the same member ID. By default, the Virtual Chassis applies configuration items that are not member-specific to all member switches.

**NOTE:** When you add or delete member switches in a Virtual Chassis configuration, internal routing changes might cause temporary traffic loss for a few seconds. Also, if removing a member switch changes a mixed Virtual Chassis into a non-mixed Virtual Chassis, you must remove the mixed mode setting on all member switches of the Virtual Chassis and reboot the Virtual Chassis; network services are disrupted until the Virtual Chassis is up again.

This topic does not apply to:

- A Virtual Chassis Fabric (VCF).

Instead, see [Removing a Device From a Virtual Chassis Fabric](#) for VCF information.

- A mixed Virtual Chassis that contains EX4200, EX4500, or EX4550 switches.

Instead, see [Removing an EX4200, EX4500, or EX4550 Switch From a Mixed Virtual Chassis \(CLI Procedure\)](#).

- An EX8200 Virtual Chassis.

To remove or replace a member switch of any other EX Series or QFX Series Virtual Chassis, use one of the following procedures that matches what you want to do.

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

To remove a switch from a Virtual Chassis without replacing it:

1. Power off and disconnect the member switch you want to remove from the Virtual Chassis.
2. If the Virtual Chassis configuration is preprovisioned, on the Virtual Chassis primary, remove the removed switch's member setting from the preprovisioned configuration.

```
[edit virtual-chassis]
user@vc-primary# delete member removed-member-id
```

If the Virtual Chassis configuration is nonprovisioned, change the mastership-priority values of each member switch as needed to reconfigure the Virtual Chassis roles. See [Configuring Primary Role of a Virtual Chassis](#).

3. (Optional) If removing a member switch leaves only two remaining member switches in the Virtual Chassis, we recommend you disable split detection in a two-member Virtual Chassis. See [Disabling Split and Merge in a Virtual Chassis](#) for details.
4. Commit any configuration changes made in previous steps.
5. (For a mixed Virtual Chassis only) If removing this member switch changes the Virtual Chassis from a mixed to a non-mixed Virtual Chassis, you must also remove the mixed-mode setting from the Virtual Chassis. See [Understanding Mixed EX Series and QFX Series Virtual Chassis](#) for details on the combinations of switches that comprise a mixed Virtual Chassis. You must reboot the Virtual Chassis for the mode change to take effect. To do this, you can include the reboot option with the `request virtual-chassis mode mixed disable` command that turns off mixed mode, or reboot all member switches of the Virtual Chassis separately when ready to do so, as shown below.

```
user@vc-primary> request virtual-chassis mode mixed disable all-members
user@vc-primary> request system reboot all-members
```

**NOTE:** Step 7 describes how to remove the mixed mode and other settings from the removed switch if needed.

6. When you remove a member switch from a Virtual Chassis configuration, the primary keeps that member switch's member ID in reserve. Use the following command on the Virtual Chassis primary to make that member ID available for reassignment:

```
user@vc-primary> request virtual-chassis recycle member-id member-id
```

7. If you want to use the removed switch as a standalone switch, you must remove any Virtual Chassis configuration items and settings on that switch. For a smooth transition to a new role as a standalone switch, we recommend to revert the switch to its default factory configuration using the [request system zeroize](#) command, and then apply the configuration items you want on the switch.

If you do not want to revert to default factory settings, use commands such as the following to remove Virtual Chassis settings for the mode and VCPs from the removed switch:

- a. If you removed the switch from a mixed-mode Virtual Chassis that is not an EX4300 mixed Virtual Chassis, disable the mixed-mode setting on the switch as follows:

```
user@switch> request virtual-chassis mode mixed disable
```

If you removed a non-multigigabit model EX4300 switch from a mixed EX4300 Virtual Chassis that contains multigigabit EX4300 switches (EX4300-48MP), when you disable mixed mode, you must also disable the special `ieee-clause-82` port mode on the removed switch if you want to reconfigure it as a standalone switch or use it in any other type of mixed Virtual Chassis or non-mixed Virtual Chassis. Otherwise, the VCPs on the switch will not connect with other Virtual Chassis members or those ports will not operate properly as network ports.

In this case, to disable mixed mode and the port mode on the switch:

```
user@switch> request virtual-chassis mode mixed ieee-clause-82 disable
```

See [Understanding EX4300 Multigigabit and Other EX4300 Model Switches in a Mixed EX4300 Virtual Chassis](#) for more information about this special port mode on EX4300 switches.

- b. Delete the VCP settings for any ports that were used as VCPs:

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

- c. Reboot the standalone switch for settings such as mode changes to take effect.

## Remove, Repair, and Reinstall the Same Switch

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

To remove, repair, and reinstall the member switch:

1. Power off and disconnect the member switch to be repaired.

2. Repair, as necessary.
3. Reconnect the switch and power it on.

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

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

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

1. Power off and disconnect the member switch to be replaced.

**NOTE:** See Step 7 in ["Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch"](#) on page 277 for information on how to disable Virtual Chassis settings from the removed switch if you want to use that switch in a different configuration.

2. If the replacement member switch has been previously configured, revert that switch's configuration to the factory defaults. See the [request system zeroize](#) command.

The replacement member switch should be powered on and running with the factory default configuration at the end of this step.

3. (Recommended for a QFX5100 Virtual Chassis under certain conditions) When you add or replace a QFX5100-24Q switch that is configured in the Routing Engine role in a QFX5100 Virtual Chassis, if the new switch has two EX4600-EM-8F expansion modules, we recommend that you set the primary role priorities on the routing engine members and the new switch to prevent a primary-role switchover to the new switch until after the new switch is completely initialized in the Virtual Chassis.

Before interconnecting the new switch into the Virtual Chassis in this case, see [Add or Replace a QFX5100-24Q Switch with Two Expansion Modules in a QFX5100 Virtual Chassis](#) for details on why, when, and how you should do this step.

**NOTE:** You might need to do this even if the new switch has the default factory configuration.

4. (Required when automatic software update is not enabled on the Virtual Chassis and the new member switch is running a version of Junos OS that is different than the version of Junos OS running on the Virtual Chassis) Ensure that the correct version of Junos OS is or will be installed on the new member switch by performing *either* of the following tasks:

- Enable automatic software update on the Virtual Chassis. See [Configuring Automatic Software Update on Virtual Chassis Member Switches](#). The Virtual Chassis will automatically update the software on the replacement switch in a later step when it is interconnected and recognized as part of the Virtual Chassis. The replacement switch does not require any action in this case for this step.
- Install the version of Junos OS running on the Virtual Chassis onto the new member switch before interconnecting it into the Virtual Chassis. For EX series switches, see [Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine \(CLI Procedure\)](#), or for QFX Series switches, see [Software Installation and Upgrade Overview](#) and [Installing Software Packages on QFX Series Devices](#). In this case, at the end of this step, the replacement switch will be running with the new version of the software and should have the factory default configuration.



**CAUTION:** You can only set up a QFX5110 Virtual Chassis with both QFX5110 and QFX5100 switches if they are running the same Junos OS image that includes “-qfx-5e-” in the software package filename (from the Junos OS Software Center). If the switch you are replacing in a QFX5110 Virtual Chassis is a QFX5100 switch that you previously installed with a “-qfx-5-” Junos OS image file, you *must* upgrade the replacement switch to a “-qfx-5e-” image instead before inserting it into the QFX5110 Virtual Chassis. See [Upgrading a QFX5100 Switch with a USB Device to Join a QFX5110 Virtual Chassis or Virtual Chassis Fabric](#). The automatic software update feature can’t update a “-qfx-5-” image to a “-qfx-5e-” image.

5. Connect one link from the replacement member switch to the Virtual Chassis as follows, depending on which type of ports you are using:
  - If you are interconnecting a switch using dedicated Virtual Chassis Ports (VCPs), connect one dedicated VCP on the replacement member switch to a dedicated VCP on another member switch in the Virtual Chassis.
  - If you are interconnecting a switch using optical ports configured as VCPs:

On the replacement switch, configure the optical ports that you are using to connect to the Virtual Chassis as VCPs. (You should also configure the optical ports on the existing member switches in the Virtual Chassis where the replacement member switch will be connected, if they are not already configured.) To configure an optical port as a VCP:

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

Connect one configured optical port VCP on the replacement switch to a configured optical port VCP on another member switch in the Virtual Chassis.

**NOTE:** You can set optical port VCPs on a standalone switch before interconnecting one link into an existing Virtual Chassis, or set them after connecting the link. In either case, you must set the ports as VCPs for the primary to detect and complete the process of adding the switch as a member. For more information on setting up VCPs on EX Series switches, see [Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port](#), and for details on which ports can be configured as VCPs in a QFX Series Virtual Chassis, see [Understanding Virtual Chassis Components](#).

6. Confirm that the new member switch is now included in the Virtual Chassis configuration on switches with a front-panel LCD by checking the display for the member ID. It should show a member ID in the range from 0 through 9.

If you are using a switch that does not have an LCD interface, enter the `show virtual-chassis` command and view the output to confirm the switch is part of the Virtual Chassis configuration.

7. Cable the other VCP on the replacement member switch into the Virtual Chassis based on how you planned to interconnect the switch in Step 5 of this procedure.



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

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

8. IF you need to update the new member switch's current member ID to the member ID of the switch that was removed from the Virtual Chassis configuration:

- In a nonprovisioned Virtual Chassis, issue the `request virtual-chassis renumber` command on the primary member switch.
- In a preprovisioned Virtual Chassis, on the primary member switch, reconfigure the member information for the new member switch using the `[edit virtual-chassis] member` configuration statement.

To use the same member ID as the member that was replaced, associate the new switch's serial number (on the back of the switch) with the replaced member ID, as follows:

```
[edit virtual-chassis]
user@switch# set member replaced-member-ID serial-number new-member-serial-number
```

**NOTE:** You can alternatively use the `replace` configuration editing command to substitute the serial number of the replacement member switch for the replaced member's serial number in the existing configuration item for the replaced member.

To configure the new member switch with a different member ID, associate the new switch's serial number with the desired member ID and then delete the configuration item for the replaced member switch, as follows:

```
[edit virtual-chassis]
user@switch# set member new-member-ID serial-number new-member-serial-number
user@switch# delete member replaced-member-ID
```

## Replace a Member Switch With a Different Type of Switch That Changes the Virtual Chassis to Mixed Mode

If you want to replace a member switch with a different type of switch that changes the Virtual Chassis from a non-mixed to a mixed Virtual Chassis:

1. Remove the member switch as described in ["Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch" on page 277](#).
2. Use the configuration procedure for *adding a new switch to an existing Virtual Chassis* based on the type of switch you are adding. (See the list of related documentation at the bottom of this page.)

See [Understanding Mixed EX Series and QFX Series Virtual Chassis](#) for the combinations of switches that comprise a mixed Virtual Chassis.

### RELATED DOCUMENTATION

[Adding a New Switch to an Existing EX2300, EX3400, or EX4300 Virtual Chassis](#)

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

[Adding a New Switch to an Existing EX4650 or QFX Series Virtual Chassis](#)

[Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\)](#)

[Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\)](#)

[Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\)](#)

[Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\)](#)

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

## Removing an EX4200, EX4500, or EX4550 Switch From a Mixed Virtual Chassis (CLI Procedure)

This topic explains how to remove a switch from a mixed Virtual Chassis. It also explains how to reconfigure the Virtual Chassis in cases in which you are removing the switches to convert a mixed Virtual Chassis into an EX4200 Virtual Chassis, an EX4500 Virtual Chassis, or an EX4550 Virtual Chassis.

If you want to replace a member switch of a Virtual Chassis that only contains the same type of member switches, see *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*.

To remove a switch from a mixed Virtual Chassis:

1. Power off and disconnect the member switch that is being removed from the Virtual Chassis.
2. If the Virtual Chassis configuration was preprovisioned, remove the switch from the preprovisioned configuration. See *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*. If the Virtual Chassis was nonprovisioned, change the primary-role priority values of each member switch as needed to reconfigure the Virtual Chassis roles. See *Configuring Primary Role of a Virtual Chassis*.
- 3.

**NOTE:** You only have to set mixed mode when configuring an EX4200 switch into a Virtual Chassis that has EX4500 or EX4550 switches. You, therefore, do not have to have this setting unless an EX4200 switch is in your Virtual Chassis.

If you are removing the last EX4200 switch from a mixed Virtual Chassis, disable mixed Virtual Chassis mode for all of the switches in the Virtual Chassis:

```
user@switch> request virtual-chassis mode mixed disable all-members
```

Reboot all member switching the Virtual Chassis to complete this step:

```
user@switch> request system reboot all-members
```



4. If you want to place the removed switch back onto the network as a standalone switch, disable mixed Virtual Chassis mode on the switch:

```
user@switch> request virtual-chassis mode mixed disable
```

5. (EX4500 switch only) Set the PIC mode on the removed switch to **intraconnect**:

```
user@switch> request chassis pic-mode intraconnect
```

You only need to perform this step if you want to put the EX4500 switch back onto the network as a standalone switch.

6. Reboot the standalone switch so the new settings can take effect.

## Configuring Primary Role of a Virtual Chassis

### IN THIS SECTION

- [Configuring Primary Role Using a Preprovisioned Configuration File | 285](#)
- [Configuring Primary Role Using a Configuration File That Is Not Preprovisioned | 286](#)

**NOTE:** This topic applies to all QFX Virtual Chassis and all EX Series Virtual Chassis except EX8200 Virtual Chassis.

A Virtual Chassis configuration has two Routing Engines—one is the switch in the primary Routing Engine role and the other is the switch in the backup Routing Engine role. The remaining members operate in the linecard role. You can designate the role (primary, backup, or linecard) that a member switch performs within any Virtual Chassis whether or not you are using a preprovisioned configuration. For details on which switches in a mixed Virtual Chassis we recommend or require you to configure into the primary or backup Routing Engine role, see *Understanding Mixed EX Series and QFX Series Virtual Chassis*.

**NOTE:** We recommend that you always use `commit synchronize` rather than `commit` to save configuration changes for a Virtual Chassis to ensure that the changes are saved on both Routing Engines.

## Configuring Primary Role Using a Preprovisioned Configuration File

To configure primary role using a preprovisioned configuration:

1. Note the serial numbers of the switches that you want to function as the primary and backup Routing Engines.

**NOTE:** Serial number values are case-sensitive.

2. Power on only the switch that you want to function as the primary Routing Engine.
3. Edit the configuration to specify the preprovisioned configuration mode:

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

4. Specify the serial numbers of the member switches that you want to function as primary and backup, specifying their role as `routing-engine`:

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

**NOTE:** You cannot directly modify the primary-role priority value when you configure a member switch in a preprovisioned Virtual Chassis. The Virtual Chassis generates the primary-role priority values automatically. The values depend on the role you assign to the member switches in the configuration file. The default primary-role priority on any switch is 128. When you configure the primary and backup Routing Engine members, the primary-role priority changes to 129 on those members. According to the primary-role election algorithm, the member that you powered on first gets priority and becomes the primary. See *Understanding How the Primary in a Virtual Chassis Is Elected*. You can configure only two members with the `routing-engine` role.

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

## Configuring Primary Role Using a Configuration File That Is Not Preprovisioned

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

1. Power on only the switch that you want to function as the primary Routing Engine.
2. Configure the highest possible primary-role priority value (255) for the member that you want to function as the primary Routing Engine:

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

3. Configure the same primary-role priority value (continue to edit the Virtual Chassis configuration on the primary) for the member that you want to be the backup Routing Engine:

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

**NOTE:** You should assign the same (highest) primary-role priority value to both the primary and backup Routing Engine members for reliable graceful Routing Engine switchover (GRES) operation. This and the other primary-role election considerations also help keep primary role from switching back and forth rapidly between the two members under failover conditions.

4. Use the default primary-role priority value (128) for the remaining member switches or configure the primary-role priority to a value that is lower than the value specified for members acting in the primary and backup Routing Engine roles.

## RELATED DOCUMENTATION

*Understanding Mixed EX Series and QFX Series Virtual Chassis*

*Configuring an EX4650 or a QFX Series Virtual Chassis*

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

*Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)*

[Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) | 269](#)

[Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 271](#)

## Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port

### IN THIS SECTION

- [When to Configure VCPs | 288](#)
- [Prepare Virtual Chassis Member Switches Before Converting VCPs | 289](#)
- [Set Uplink Ports to Form VCP Links Between Member Switches in a Virtual Chassis | 290](#)
- [Set an Uplink Port as a VCP on a Standalone Switch | 291](#)
- [Remove a VCP Setting on an Uplink or Network Port | 292](#)

Use the procedures described in this topic to set up Virtual Chassis ports (VCPs) to connect two switches together in an EX Series or a QFX Series Virtual Chassis.

Switches that can be members of a Virtual Chassis might have:

- Dedicated VCPs—Ports you can use only as VCPs.
- Default-configured VCPs—Ports that are already configured into VCPs with the default factory configuration. On some switches these ports can alternatively be converted back into and used as uplink or network ports.
- Uplink or network ports that are also supported as VCPs—Ports you can configure into VCPs and convert back into uplink or network ports as needed.

If you don't have or are not using dedicated or default-configured VCPs, you need to convert supported ports into VCPs to interconnect Virtual Chassis members. When a switch has uplink or network ports that can be converted into VCPs, you can use some ports as VCPs and others as network ports or uplinks to other devices in trunk mode. When you set a port as a VCP, you can't use it for any other purpose.

**NOTE:** You don't use the procedures in this topic to set the VCPs on EX4400 switches. EX4400 switches have ports that operate as VCPs by default, and no other ports on the switch can be used as VCPs. If you change the operational mode of the default VCPs to network port mode to

use them as network ports instead of as VCPs, to subsequently use the switch in a Virtual Chassis, you must disable network port mode to return those ports to their default VCP mode. To change the default VCPs to network port mode, use the `request virtual-chassis mode network-port <reboot>` command. To disable network port mode to convert the ports back into VCPs, use the `request virtual-chassis mode network-port disable <reboot>` command. (When you enable or disable network port mode, you must reboot the switch for the change to take effect.)

You can set ports as VCPs on a standalone switch before interconnecting any links into a Virtual Chassis, or set them after interconnecting one link on the switch into an existing Virtual Chassis. Either way, after the VCP is connected into the Virtual Chassis, the primary switch uses the link to detect the switch and complete the process of adding it as a member.

**NOTE:** Most Virtual Chassis also support *autoprovisioning*, which means that under certain conditions, when you interconnect a member switch into an existing Virtual Chassis, ports that are supported as VCPs will convert automatically into VCPs when you cable the link. This is an easy way to add member switches to a Virtual Chassis without needing to explicitly configure VCPs, but it only works if the ports on *both* sides of the link are *not already configured as VCPs*. If you want to use autoprovisioning, you might need to delete VCP settings (whether default-configured or those you previously set explicitly) on either or both sides of the links you are using to interconnect the new member switch. See *Automatic Virtual Chassis Port (VCP) Conversion* for details.

See *Virtual Chassis Port Options* for a list of supported VCP ports on each type of switch. For complete details about where a switch has dedicated VCPs, default-configured VCPs, or ports that can be configured as VCPs, and what transceivers and cables are supported to use for VCP connections on that switch, see the hardware documentation for the switch. In general, even if a port is supported as a VCP, you can't use it as a VCP if it's channelized.

## When to Configure VCPs

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

- You are configuring a Virtual Chassis composed of switches that support Virtual Chassis but do not have default-configured VCPs or dedicated VCPs.
- You are using default-configured VCPs or dedicated VCPs to interconnect members in a Virtual Chassis, and want to add redundant VCP links between members using additional ports that can be configured into VCPs.

- You want to interconnect two switches into a Virtual Chassis that have dedicated VCPs but are located in different wiring closets or sites, and the switches are farther apart than the maximum length of a dedicated VCP cable.
- You previously changed a default-configured VCP to use it as a network or uplink port, and now you want to use it as a VCP again.

We recommend that you have two uplink VCP connections within each wiring closet for redundancy. VCPs automatically bundle into a Link Aggregation Group (LAG) when two or more ports operating at the same speed are configured into VCPs between the same two member switches. See *Understanding Virtual Chassis Port Link Aggregation* for details.



**CAUTION:** If you set a port as a VCP to create a redundant link with a dedicated VCP connection on EX4200, EX4500, or EX4550 switches, to avoid traffic looping within the Virtual Chassis, you must reboot the Virtual Chassis after configuring the port conversion and cabling the VCP link. See *Troubleshooting an EX Series Virtual Chassis* for more information.

## Prepare Virtual Chassis Member Switches Before Converting VCPs

Before converting a port into a VCP and interconnecting the switch into a Virtual Chassis:

1. Verify which ports can be used as VCPs in your particular configuration. See *Virtual Chassis Port Options* for a summary of the VCP options on switches that support Virtual Chassis, and the hardware documentation for each type of switch for complete details about the ports and installed transceivers that can be used as VCPs.
2. If you are configuring an uplink module port as a VCP, if needed, install the uplink module in the member switches that you want to interconnect.
3. Log into the switch that is or will be the primary of the Virtual Chassis.

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

4. (EX Series switches only) Run EZSetup on the switch that you are configuring to be the primary member switch. Make sure the hostname and other identification, time zone, and network properties are set up on the primary. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details. The parameters you specify for the primary apply to the entire Virtual Chassis, including all the member switches that you interconnect later.
5. If you want to configure and manage the Virtual Chassis remotely, specify the VME global management interface. You can configure the VME global management interface when you are setting up the primary or you can do it after completing the other configuration steps for the Virtual Chassis. See ["Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\)" on page 295.](#)

6. Configure primary role of the Virtual Chassis using either a nonprovisioned or preprovisioned configuration. See *Configuring Primary Role of a Virtual Chassis* for details.

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

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

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

## Set Uplink Ports to Form VCP Links Between Member Switches in a Virtual Chassis

From the Virtual Chassis, you can set an uplink port on the local member or on a specified member as a VCP.

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

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

1. Set one uplink port of member 0 as a VCP. You don't need to specify the member *member-id* option, because the command applies by default on the member where it is executed. You can alternatively include the `local` option if you want to make sure the command applies only to that port locally on the switch where you're running the command.

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

2. Set one uplink port of member 1 as a VCP. This step includes the member *member-id* option because it acts on a different member switch than the local member switch.

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

**NOTE:** You can also connect to a member switch individually using the *request session member* command and set a VCP locally on that member. (You don't specify the *member* option in that case.)

## Set an Uplink Port as a VCP on a Standalone Switch

You can set an uplink VCP on a standalone switch before interconnecting the link into an existing Virtual Chassis. You must set the port as a VCP for the Virtual Chassis primary to detect the switch and complete the process of adding it as a member.

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

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

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

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

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

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



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

4. Repeat the steps above to interconnect the new switch to another member of the Virtual Chassis or to create redundant VCP links to the same member.

## Remove a VCP Setting on an Uplink or Network Port

You might want to convert a VCP back into an uplink or network port if:

- You want to use a default-configured VCP on a switch as a network or uplink port instead of as a VCP.
- You want to add a new member switch or a new VCP link in an existing Virtual Chassis using autoprovisioning, where the VCP links form automatically when you cable them *only* if the ports on both sides of the link are not already set as VCPs. (See *Automatic Virtual Chassis Port (VCP) Conversion* for details.)
- You remove a switch from a Virtual Chassis and want to use it as a standalone switch again.

**NOTE:** In this case, whenever possible we recommend that you revert the switch back to its default factory configuration to guarantee the smoothest transition back to standalone operation. See *Removing or Replacing a Member Switch of a Virtual Chassis Configuration*.

To remove the VCP setting on a port:

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

You usually do this on the switch with the VCP itself, so you don't need to include the member *member-id* option because the command applies by default on the member where you run it.

## RELATED DOCUMENTATION

[Configuring an EX3300 Virtual Chassis \(CLI Procedure\) | 240](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

[Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)

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

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

[Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets | 148](#)

[Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File | 168](#)

[Removing or Replacing a Member Switch of a Virtual Chassis Configuration](#)

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

## Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch (CLI Procedure)

You can set any optical port on an EX4500 switch as a Virtual Chassis port (VCP). Setting an optical port as a VCP is especially useful in cases in which you have to connect Virtual Chassis members together across a distance that is greater than the longest VCP cable can breach.

You can use the optical ports to connect EX4200 switches, EX4500 switches, and EX4550 switches together into a Virtual Chassis. All supported optical connections between EX4200, EX4500, and EX4550 switches can be configured as VCPs.

**NOTE:** The Virtual Chassis module for EX4500 switches is a field-replacable unit (FRU) that is not shipped with all EX4500 switch models but can be purchased separately. See [Virtual Chassis Module in EX4500 Switches](#) and [EX4500 Switch Models](#).



**CAUTION:** If you configure an uplink port as a VCP to create a redundant link with a dedicated VCP connection on EX4200, EX4500, or EX4550 switches, to avoid traffic looping within the Virtual Chassis, you must reboot the Virtual Chassis after configuring the port conversion and cabling the VCP link. See *Troubleshooting an EX Series Virtual Chassis* for more information.

Before you set an uplink port as a VCP on an EX4500 or EX4550 switch:

- Cable the uplink port connections.

To set the optical connection as a VCP:

1. Set the 10-Gigabit Ethernet SFP+ port as a VCP interface:

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

where:

- *pic-slot-number*—The PIC slot number, which is **0** when specifying a native port, **1** when specifying a port on the uplink module in PIC slot number 1, and **2** when specifying a port on an uplink module in PIC slot number 2.
- *port-number*—The port number on the switch or uplink module.

2. Log onto the other EX4500 switch, and set the other end of the connection as a VCP interface:

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

## RELATED DOCUMENTATION

[Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets](#) | 159

## Setting an Uplink Port as a Virtual Chassis Port Using the LCD Panel

This topic describes how to set or delete the uplink ports as VCPs using the LCD panel.

The following procedure shows how to configure uplink module port **ge-0/1/2** as a VCP.

To set an uplink port as a VCP using the LCD panel:

1. Press **Menu** until you see **MAINTENANCE MENU**.
2. Press **Menu** until you see **REQUEST VC PORT**.
3. Press **Enter**. You will see **SET VC PORT?**.
4. Press **Enter**. You will see **SET FPC 0?**.
5. Press **Enter**. You will see **SET PIC 0?**.
6. Press **Menu** until you see **SET PIC 1?**.
7. Press **Enter**. You will see **SET PORT 0?**.
8. Press **Menu** until you see **SET PORT 2?**.
9. Press **Enter**. You will see **CONFIGURING ...**.
10. Once the configuration has been accepted, press **Enter** to return to the **MAINTENANCE** menu.

You can also use the LCD panel to delete a VCP, thus resetting the port to an uplink port.

To reset **vcp-0/1/2** to an uplink port using the LCD panel:

1. Press **Menu** until you see **MAINTENANCE MENU**.
2. Press **Menu** until you see **REQUEST VC PORT**.
3. Press **Enter**. You will see **SET VC PORT?**.
4. Press **Menu**. You will see **DELETE VC PORT?**.
5. Press **Enter**. You will see **DELETE FPC 0?**.
6. Press **Enter**. You will see **DELETE PIC 0?**.
7. Press **Menu** until you see **DELETE PIC 1?**.
8. Press **Enter**. You will see **DELETE PORT 0?**.
9. Press **Menu** until you see **DELETE PORT 2?**.
10. Press **Enter**. You will see **CONFIGURING ....**
11. Once the configuration has been accepted, press **Enter** to return to the **MAINTENANCE** menu.

## RELATED DOCUMENTATION

[LCD Panel in EX3300 Switches](#)

[LCD Panel in EX4200 Switches](#)

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

[Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port](#)

[Understanding Interface Naming Conventions](#)

## Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis (CLI Procedure)

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

If you want to configure and manage the Virtual Chassis remotely through SSH or Telnet, configure the virtual management Ethernet (VME) interface on the primary of the Virtual Chassis. You can configure and manage all members of the Virtual Chassis through this single global interface by following these steps:

1. Power on the switch that you want to function as the primary.
2. Check the front-panel LCD to confirm that the switch has powered on correctly.
3. Run the EZSetup program on the switch, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)*.

To configure the VME interface:

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

## RELATED DOCUMENTATION

[Example: Configuring an EX3300 Virtual Chassis with a Primary and Backup | 91](#)

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

*Understanding Global Management of a Virtual Chassis*

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

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

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

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

You can disable the MAC persistence timer starting in Junos OS Release 12.1R3. When the MAC persistence timer is disabled, the MAC address of the old primary switch is used as the system MAC base address; no MAC address changes occur within the Virtual Chassis even when the old primary

switch is no longer a member of the Virtual Chassis because it has been physically disconnected or removed.

To configure or modify the MAC persistence timer:

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

To disable the MAC persistence timer:

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

## RELATED DOCUMENTATION

*Configuring an EX4650 or a QFX Series Virtual Chassis*

*Understanding Virtual Chassis Components*

## Configuring Fast Failover in an EX Series Virtual Chassis

**NOTE:** This topic does not apply to EX2200 Virtual Chassis and EX8200 Virtual Chassis.

The Virtual Chassis fast failover feature is a hardware-assisted failover mechanism that automatically reroutes traffic and reduces traffic loss in the event of a link or switch failure. If a link between two members fails, traffic flow between those members must be rerouted quickly so that there is minimal traffic loss.

Fast failover is enabled by default on dedicated Virtual Chassis ports (VCPs). You must manually enable fast failover on uplink ports that have been configured as VCPs or on EX3300 uplink ports that are configured as VCPs by default.

Before you begin configuring fast failover, ensure that the dedicated VCPs or uplink VCPs are connected in a ring topology.

- To reenable the fast failover feature on all dedicated VCPs in a ring:

```
[edit]
user@switch# delete virtual-chassis fast-failover vcp disable
```

- To configure the fast failover feature on all XFP uplink VCPs in a ring:

```
[edit]
user@switch# set virtual-chassis fast-failover xe
```

- To configure the fast failover feature on all SFP uplink VCPs in a ring:

```
[edit]
user@switch# set virtual-chassis fast-failover ge
```

## RELATED DOCUMENTATION

[Understanding Fast Failover in an EX Series Virtual Chassis | 57](#)

[Disabling Fast Failover in an EX Series Virtual Chassis | 298](#)

[Understanding Fast Failover in an EX Series Virtual Chassis | 57](#)

## Disabling Fast Failover in an EX Series Virtual Chassis

**NOTE:** This topic does not apply to EX2200 Virtual Chassis and EX8200 Virtual Chassis.

Fast failover is enabled by default on dedicated Virtual Chassis ports (VCPs). You must use this procedure to manually disable fast failover on dedicated VCPs or to disable fast failover on uplink ports that you previously configured as VCPs.

- To disable the fast failover feature on all dedicated VCPs in a ring:

```
[edit]
user@switch# set virtual-chassis fast-failover vcp disable
```

- To disable the fast failover feature on all XFP uplink VCPs in a ring:

```
[edit]
user@switch# delete virtual-chassis fast-failover xe
```

- To disable the fast failover feature on all SFP uplink VCPs in a ring:

```
[edit]
user@switch# delete virtual-chassis fast-failover ge
```

## RELATED DOCUMENTATION

[Understanding Fast Failover in an EX Series Virtual Chassis | 57](#)

[Configuring Fast Failover in an EX Series Virtual Chassis | 297](#)

[Understanding Fast Failover in an EX Series Virtual Chassis | 57](#)

## Disabling Split and Merge in a Virtual Chassis

The split and merge feature is enabled by default on all EX Series switches and QFX Series devices that are connected into a Virtual Chassis.

If a Virtual Chassis splits:

- Each part detects the members they can still reach and elects a new primary for those members to form a smaller Virtual Chassis.
- Each part determines if it will remain the active new configuration of the original Virtual Chassis. The feature's rules ensure only one part remains active after a split.

See *Understanding Split and Merge in a Virtual Chassis* for complete details on what happens during a Virtual Chassis split or merge.

You can disable the split and merge feature in a Virtual Chassis by configuring the `no-split-detection` option on the Virtual Chassis. In this case, if the Virtual Chassis splits, both parts of the split Virtual Chassis configuration usually remain active.

However, with split detection disabled, be aware that the two resulting Virtual Chassis might not form as you might expect. For example, in a preprovisioned Virtual Chassis:



- If both of the Routing Engines end up in the same Virtual Chassis configuration after a split, the other part of the split Virtual Chassis configuration can't form and remains inactive (because it has no members that can take on the primary or backup Routing Engine).
- If the Routing Engines end up in different parts of the split Virtual Chassis configuration and the rest of the member switches are configured in linecard roles, then the resulting Virtual Chassis parts might not be able to select a backup Routing Engine.

**BEST PRACTICE:** We very strongly recommend that you:

- Keep the default setting with the split and merge feature enabled for any Virtual Chassis with more than two members.

This provides the most predictable results when a split happens, and enables the split and merge feature to perform a merge after the issues that caused the split are resolved.

- Disable split and merge on a Virtual Chassis with only two member switches by setting the `no-split-detection` option.

We have found that a two-member switch Virtual Chassis with split and merge disabled reforms more quickly with fewer complications when a split occurs.

If you expand a two-member Virtual Chassis to include more members, delete the `no-split-detection` setting to re-enable split and merge feature again.

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

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

To remove this configuration item to restore the default configuration with the split and merge feature again—we very strongly recommend that you do this when a two-member Virtual Chassis is expanded to have more members:

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

## RELATED DOCUMENTATION

*Understanding Split and Merge in a Virtual Chassis*

## Configuring Automatic Software Update on Virtual Chassis Member Switches

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

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

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

To configure the automatic software update feature for an EX Series or QFX Series Virtual Chassis that is not a mixed-mode Virtual Chassis:

[edit]

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

**NOTE:** An EX4300 Virtual Chassis with a combination of EX4300 multigigabit model switches (EX4300-48MP) and other EX4300 switches must be configured in mixed mode, and the automatic software update feature is not supported in this case.

**NOTE:** A QFX5110 Virtual Chassis is considered to be a non-mixed Virtual Chassis. Because both QFX5110 and QFX5100 switches in a QFX5110 Virtual Chassis run the same software image, you can use the `auto-sw-update` command with one software package name for all members.



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

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

To configure the automatic software update feature on a mixed Virtual Chassis containing at least one EX4200 switch and at least one EX4500 or EX4550 switch, use the `ex-4200` option when you are specifying a path to a package for the EX4200 switches and the `ex-4500` option when you are specifying a path to a package for the EX4500 or EX4550 switches, as follows:

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

You do not need to specify the `ex4500` option when configuring automatic software update for a Virtual Chassis with only a combination of EX4500 and EX4550 switches, which does not need to run in mixed mode. Specifying only a Junos OS package name for an EX4500 switch updates the software for both EX4500 and EX4550 switches.

To configure the automatic software update feature on a mixed QFX5100 Virtual Chassis composed of QFX5100 switches and at least one other supported type of device (QFX3500, QFX3600, or EX4300 switches), you must specify a software package name for each type or family of device in the mixed Virtual Chassis:

- Specify the `qfx-5` option with the path to the Junos OS package for QFX5100 switches
- Specify the `qfx-3` option with the path to the Junos OS package for QFX3600 and QFX3500 switches
- Specify the `ex4300` option with the path to the Junos OS package for EX4300 switches

as follows:

```
[edit]
user@device# set virtual-chassis auto-sw-update qfx-5 package-name package-name
user@device# set virtual-chassis auto-sw-update qfx-3 package-name package-name
user@device# set virtual-chassis auto-sw-update ex-4300 package-name package-name
```

You can similarly specify different package names using the `ex-4600` and `ex-4300` options with the `auto-sw-update` statement to configure the automatic software update feature for a mixed EX4600 Virtual Chassis that contains EX4600 and EX4300 member switches.

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

*/pathname/package-name*

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

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

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

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

## RELATED DOCUMENTATION

*Understanding Automatic Software Update on Virtual Chassis Member Switches*

*Understanding Mixed EX Series and QFX Series Virtual Chassis*

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

### IN THIS SECTION

- [Preparing the Switch for Software Installation | 304](#)
- [Upgrading the Software Using NSSU | 305](#)

You can use nonstop software upgrade (NSSU) to upgrade the software running on all member switches in most EX Series Virtual Chassis with minimal traffic disruption during the upgrade.

NSSU is supported on the following EX Series Virtual Chassis platforms:

- EX3300 Virtual Chassis
- EX3400 Virtual Chassis
- EX4200 Virtual Chassis
- EX4300 Virtual Chassis

- EX4500 Virtual Chassis
- EX4550 Virtual Chassis
- All mixed Virtual Chassis composed of EX4200, EX4500, and EX4550 switches
- EX8200 Virtual Chassis

This topic covers:

## Preparing the Switch for Software Installation

Before you begin software installation using NSSU:

- Ensure that the Virtual Chassis is configured correctly to support NSSU. Verify that:
  - The Virtual Chassis members are connected in a ring topology. A ring topology prevents the Virtual Chassis from splitting during an NSSU.
  - The Virtual Chassis primary and backup are 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 is preprovisioned so that the linecard role has been explicitly assigned to member switches acting in the 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 other member switches must maintain their linecard roles.

For information on configuring a preprovisioned Virtual Chassis, see ["Configuring an EX3300 Virtual Chassis \(CLI Procedure\)" on page 240](#), ["Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)" on page 245](#), *Configuring an EX2300, EX3400, or EX4300 Virtual Chassis*, and *Configuring an EX8200 Virtual Chassis (CLI Procedure)*.

- A two-member Virtual Chassis has `no-split-detection` configured so that the Virtual Chassis does not split when an NSSU upgrades a member.
- Verify that the members are running the same version of the software:

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

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.

- 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:

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

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

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

- 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 may split during the upgrade. A split Virtual Chassis can cause disruptions to your network, and you may have to manually reconfigure your Virtual Chassis after the NSSU if the split and merge feature was disabled. For more information about a split Virtual Chassis, see *Understanding Split and Merge in a Virtual Chassis*.
- (Optional) Enable nonstop bridging (NSB). Enabling NSB ensures that all NSB-supported Layer 2 protocols operate seamlessly during the Routing Engine switchover that is part of the NSSU.
- (Optional) Back up the system software—Junos OS, the active configuration, and log files—on each member to an external storage device with the `request system snapshot` command.

## Upgrading the Software Using NSSU

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

To upgrade all members using NSSU:

1. Download the software package. If you are upgrading the software running on a mixed Virtual Chassis, download the software packages for both switch types.
2. Copy the software package or packages to the Virtual Chassis. We recommend that you copy the file to the `/var/tmp` directory on the primary.
3. Log in to the Virtual Chassis using the console connection or the virtual management Ethernet (VME) interface. Using a console connection allows you to monitor the progress of the primary switch reboot.

#### 4. Start the NSSU:

- On an EX3300 Virtual Chassis, EX3400 Virtual Chassis, EX4200 Virtual Chassis, EX4300 Virtual Chassis, EX4500 Virtual Chassis, or EX4550 Virtual Chassis, enter:

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

where *package-name.tgz* is, for example, *jinstall-ex4200-12.1R2.5-domestic-signed.tgz*.

- On a mixed Virtual Chassis, enter:

```
user@switch> request system software nonstop-upgrade set [/var/tmp/package-  
name.tgz /var/tmp/package-name.tgz]
```

where *[/var/tmp/package-name.tgz /var/tmp/package-name.tgz]* specifies the EX4200 and EX4500 software packages.

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

```
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

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)
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 user@switch ***

System going down IMMEDIATELY

Shutdown NOW!
[pid 9336]

```

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

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

6. To ensure that the resilient dual-root partitions feature operates correctly, copy the new Junos OS image into the alternate root partitions of all members:

```
user@switch> request system snapshot slice alternate all-members
```

Resilient dual-root partitions allow the switch to boot transparently from the alternate root partition if the system fails to boot from the primary root partition.



## RELATED DOCUMENTATION

*Understanding Nonstop Software Upgrade on EX Series Switches*

[Configuring Dual-Root Partitions](#)

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

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

*Understanding Nonstop Software Upgrade on EX Series Switches*

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

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

To configure the Virtual Chassis ID:

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

## RELATED DOCUMENTATION

*Understanding Split and Merge in a Virtual Chassis*

## Configuring Graceful Routing Engine Switchover in a Virtual Chassis

In a Virtual Chassis, one member switch is assigned the primary role and has the primary Routing Engine. Another member switch is assigned the backup role and has the backup Routing Engine. Graceful Routing Engine switchover (GRES) enables the primary and backup Routing Engines in a Virtual Chassis configuration to switch from the primary to backup without interruption to packet forwarding as a hitless failover solution. When you configure graceful Routing Engine switchover, the backup Routing Engine automatically synchronizes with the primary Routing Engine to preserve kernel state information and the forwarding state.

To set up the Virtual Chassis configuration to use graceful Routing Engine switchover (GRES):

1. Set up a minimum of two switches in a Virtual Chassis configuration with primary-role priority of 255:

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

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

2. Set up graceful Routing Engine switchover:

```
[edit]
user@switch# set chassis redundancy graceful-switchover
```

Commit the configuration.

**NOTE:** We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis.

## RELATED DOCUMENTATION

[Example: Configuring an EX4200 Virtual Chassis with a Primary and Backup in a Single Wiring Closet | 98](#)

[High Availability Features for EX Series Switches Overview](#)

[Understanding EX Series Virtual Chassis](#)

[Understanding QFX Series Virtual Chassis](#)

## Resetting the EX4500 Switch to Run as a Standalone Switch (CLI Procedure)

If you want to use an EX4500 switch as a standalone switch, we recommend that you set the PIC mode as `intraconnect`. The PIC mode setting determines whether the module that is plugged into the EX4500 switch operates as an `intraconnect` module or a Virtual Chassis module.

**NOTE:** Operating an EX4500 switch without the intraconnect module or the Virtual Chassis module is not supported.

**NOTE:** If you use the EX4500 switch as a standalone switch without changing the PIC mode setting to intraconnect, you will not be able to achieve 960 Gbps data rate throughput. Some of the bandwidth is reserved for the dedicated Virtual Chassis ports (VCPs).

The PIC mode is set to virtual-chassis in the factory default configuration. You cannot change the PIC mode setting in the factory default configuration; you must commit some other configuration change and then change the PIC mode setting.

To reset the PIC mode so that you can use the switch as a standalone switch:

1. Make any change to the default configuration file. For example, you could change the hostname:

```
[edit]
user@switch# set system host-name solo-4500
```

2. Commit the updated configuration file, so that it is no longer the factory default configuration:

```
[edit]
user@switch# commit
```

3. Change the setting of the PIC mode:

```
user@switch> request chassis pic-mode intraconnect
```

4. Reboot the switch to apply this changed PIC mode setting:

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

5. Verify that the PIC mode setting has been changed to intraconnect:

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

```
fpc0:
```

```
-----  
Pic Mode: Intraconnect
```

## RELATED DOCUMENTATION

[Verifying the Setting for the PIC Mode on an EX4500 Switch in a Virtual Chassis | 371](#)

[EX4500 Default Configuration](#)

## Enabling VLAN Pruning for Broadcast, Multicast, and Unknown Unicast Traffic in an EX Series Virtual Chassis (CLI Procedure)

You can enable VLAN pruning for VLANs assigned to interfaces in an EX Series Virtual Chassis. When you enable VLAN pruning for a VLAN in a Virtual Chassis, all broadcast, multicast, and unknown unicast traffic entering that VLAN uses the shortest possible path through the Virtual Chassis to the egress VLAN interface. Enabling VLAN pruning allows you to conserve bandwidth within the Virtual Chassis, since all broadcast, multicast, and unknown unicast traffic in a VLAN is broadcast to all Virtual Chassis member switches when VLAN pruning is disabled.

**BEST PRACTICE:** We recommend enabling VLAN pruning when configuring a VLAN on an EX Series Virtual Chassis.

To enable VLAN pruning when configuring a VLAN:

```
[edit]  
user@switch# set vlans vlan-name vlan-prune
```

## RELATED DOCUMENTATION

[Configuring VLANs for EX Series Switches](#)

# Configuration Statements

## IN THIS CHAPTER

- [auto-conversion](#) | 313
- [auto-sw-update](#) | 315
- [fast-failover](#) | 319
- [graceful-restart \(Enabling Globally\)](#) | 320
- [graceful-switchover](#) | 323
- [id](#) | 324
- [location \(Virtual Chassis\)](#) | 325
- [mac-persistence-timer](#) | 327
- [mastership-priority](#) | 329
- [member](#) | 331
- [no-management-vlan](#) | 333
- [no-split-detection](#) | 334
- [preprovisioned](#) | 336
- [redundancy \(Graceful Switchover\)](#) | 337
- [role](#) | 339
- [serial-number](#) | 342
- [traceoptions \(Virtual Chassis\)](#) | 343
- [vc-port](#) | 347
- [vcp-snmp-statistics](#) | 348
- [virtual-chassis](#) | 350

## auto-conversion

### IN THIS SECTION

- [Syntax | 313](#)
- [Hierarchy Level | 313](#)
- [Description | 313](#)
- [Required Privilege Level | 314](#)
- [Release Information | 314](#)

### Syntax

```
auto-conversion;
```

### Hierarchy Level

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

### Description

Enable automatic Virtual Chassis port (VCP) conversion (also called VCP autoprovisioning) in a Virtual Chassis.

**NOTE:** This statement is available only for EX Series switches in a Virtual Chassis that have automatic VCP conversion *disabled* by default—EX2200, EX3300, EX4200, EX4500, and EX4550 switches. On these EX Series switches, the automatic VCP conversion feature was *enabled* by default in Junos OS Releases prior to 14.1X53-D47 and 15.1R7. This feature is disabled by default and this statement added as of those releases to allow you to selectively enable the feature. See "[Automatic Virtual Chassis Port \(VCP\) Conversion](#)" on page 32 for details.

Configuring this statement enables the feature for the entire Virtual Chassis, and subsequently removing this statement returns the Virtual Chassis to the default behavior, in which automatic VCP conversion is disabled.

VCP links connect members in a Virtual Chassis using supported VCP ports on the member switches. Ports that are supported as VCPs and are not dedicated VCPs or configured as VCPs by default must be converted into VCPs to use them to interconnect Virtual Chassis members. You can manually convert a supported port into a VCP port using the `request virtual-chassis vc-port` command, or when automatic VCP conversion is enabled, a port will be automatically converted into a VCP under the following conditions:

- LLDP is enabled on the interfaces for the members on both sides of the link. The two interfaces exchange LLDP packets to accomplish the port conversion.
- The Virtual Chassis must be preprovisioned with the switches on both sides of the link already configured in the members list of the Virtual Chassis using the `set virtual-chassis member` command.
- The ports on both ends of the link are supported as VCPs and are *not* already configured as VCPs.

You might want to invoke automatic VCP conversion when adding a new switch to an existing Virtual Chassis or adding a redundant VCP link between two existing members of a Virtual Chassis.

**NOTE:** When automatic VCP conversion is enabled in a Virtual Chassis with switches that have dedicated VCPs (EX4200, EX4500, or EX4550 Virtual Chassis), if network or uplink ports are automatically converted into VCPs to create a redundant link with a dedicated VCP connection, you must reboot the Virtual Chassis to avoid creating a traffic loop within the Virtual Chassis. See ["Troubleshooting an EX Series Virtual Chassis" on page 504](#) for more information on this potential issue.

## 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 Releases 14.1X53-D47 and 15.1R7.

## RELATED DOCUMENTATION

[Virtual Chassis Overview for Switches | 2](#)

[Understanding Virtual Chassis Components | 21](#)

[Setting an Uplink Port on an EX Series or QFX Series Switch as a Virtual Chassis Port | 287](#)

## auto-sw-update

### IN THIS SECTION

- [Syntax | 315](#)
- [Hierarchy Level | 315](#)
- [Description | 315](#)
- [Default | 317](#)
- [Options | 317](#)
- [Required Privilege Level | 318](#)
- [Release Information | 318](#)

### Syntax

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

### Hierarchy Level

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

### Description

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

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



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

You must enter the `auto-sw-update` statement multiple times—once for each device family in your mixed Virtual Chassis or VCF—in most scenarios when enabling the automatic software update for a mixed Virtual Chassis or VCF.

The Junos OS package for an EX4500 switch updates the software for EX4500 and EX4550 switches. You do not, therefore, need to specify the `ex-4500` keyword when configuring automatic software update for a mixed Virtual Chassis that include EX4500 and EX4550 switches only. You also only have to enter the `ex-4500` keyword once to configure automatic software update for all EX4500 and EX4550 member switches in the same mixed Virtual Chassis.

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

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



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

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

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

## Default

The automatic software update feature is disabled.

## Options

**package-name**  
**package-name**

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

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

*/pathname/package-name*

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

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

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

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

**ex-4200** Specify a path to a Junos OS image for an EX4200 switch when enabling automatic software update for a mixed EX4200 and EX4500 Virtual Chassis, mixed EX4200 and EX4550 Virtual Chassis, or mixed EX4200, EX4500, or EX4550 Virtual Chassis.

**ex-4300** Specify a path to a Junos OS image for an EX4300 switch when enabling automatic software update for a mixed Virtual Chassis or VCF. Automatic software update is not supported for mixed mode EX4300 Virtual Chassis with a combination of EX4300 multigigabit model (EX4300-48MP) switches and other EX4300 switches, and this option is not applicable in that case.

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

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

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

- ex-4600** Specify a path to a Junos OS image for an EX4600 switch when enabling automatic software update for a mixed Virtual Chassis.
- qfx-3** Specify a path to a Junos OS image for a QFX3500, QFX3600, or both types of devices when enabling automatic software update for a mixed Virtual Chassis or VCF.
- qfx-5** Specify a path to a Junos OS image for a QFX5100 device when enabling automatic software update for a mixed QFX5100 Virtual Chassis or VCF. You do not need to use this option for a QFX5110 Virtual Chassis or VCF that contains both QFX5110 and QFX5100 switches because the same software image runs on either type of switch.

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

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

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

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

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

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

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

## RELATED DOCUMENTATION

[Configuring Automatic Software Update on Virtual Chassis Member Switches](#)

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

*Understanding Software Upgrades in a Virtual Chassis Fabric*

[Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches](#)

## fast-failover

### IN THIS SECTION

- [Syntax | 319](#)
- [Hierarchy Level | 319](#)
- [Description | 319](#)
- [Default | 319](#)
- [Options | 320](#)
- [Required Privilege Level | 320](#)
- [Release Information | 320](#)

### Syntax

```
fast-failover (ge | vcp disable | xe);
```

### Hierarchy Level

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

### Description

Enable the fast failover feature on an uplink Virtual Chassis port (VCP).

Fast failover is not supported on an EX2200 Virtual Chassis or EX8200 Virtual Chassis.

### Default

Fast failover is enabled, by default, on dedicated VCPs.

Fast failover is not enabled on uplink VCPs, including the uplink ports that are, by default, configured as VCPs on an EX3300 switch.

## Options

- **ge**—Enable fast failover on all Gigabit Ethernet uplink VCPs in the ring topology.
- **vcp disable**—Disable fast failover on all dedicated VCPs in the ring topology.
- **xe**—Enable fast failover on all 10-Gigabit Ethernet uplink VCPs in the ring topology.

## 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 Fast Failover in an EX Series Virtual Chassis | 57](#)

[Configuring Fast Failover in an EX Series Virtual Chassis | 297](#)

[Disabling Fast Failover in an EX Series Virtual Chassis | 298](#)

## graceful-restart (Enabling Globally)

### IN THIS SECTION

- [Syntax | 321](#)
- [Hierarchy Level | 321](#)
- [Description | 321](#)
- [Default | 322](#)
- [Options | 322](#)
- [Required Privilege Level | 322](#)
- [Release Information | 322](#)

## 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](#)

## graceful-switchover

### IN THIS SECTION

- [Syntax | 323](#)
- [Hierarchy Level | 323](#)
- [Description | 323](#)
- [Default | 323](#)
- [Required Privilege Level | 323](#)
- [Release Information | 324](#)

### Syntax

```
graceful-switchover;
```

### Hierarchy Level

```
[edit chassis (EX Series) redundancy]
```

### Description

For switches with more than one Routing Engine, including those in a Virtual Chassis or a Virtual Chassis Fabric, configure the primary Routing Engine to switch over gracefully to a backup Routing Engine without interruption to packet forwarding.

### Default

Graceful Routing Engine switchover (GRES) is disabled.

### 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 9.2.

### RELATED DOCUMENTATION

- [Example: Configuring Nonstop Active Routing on Switches](#)
- [Configuring Graceful Routing Engine Switchover](#)
- [\*Configuring Graceful Routing Engine Switchover in a Virtual Chassis\*](#)
- [Configuring Nonstop Active Routing on Switches](#)
- [Installing Software on an EX Series Switch with Redundant Routing Engines \(CLI Procedure\)](#)

## id

### IN THIS SECTION

- Syntax | 324
- Hierarchy Level | 324
- Description | 325
- Options | 325
- Required Privilege Level | 325
- Release Information | 325

## 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 | 326](#)

● [Hierarchy Level | 326](#)

- [Description | 326](#)
- [Options | 326](#)
- [Required Privilege Level | 326](#)
- [Release Information | 326](#)

## 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 | 327](#)
- [Hierarchy Level | 327](#)
- [Description | 327](#)
- [Default | 328](#)
- [Options | 328](#)
- [Required Privilege Level | 328](#)
- [Release Information | 328](#)

### 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*

## mastership-priority

### IN THIS SECTION

- [Syntax | 329](#)
- [Hierarchy Level | 329](#)
- [Description | 329](#)
- [Default | 330](#)
- [Options | 330](#)
- [Required Privilege Level | 330](#)
- [Release Information | 330](#)

### Syntax

```
mastership-priority number;
```

### Hierarchy Level

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

### Description

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

A Virtual Chassis or VCF configuration has two member switches running in the Routing Engine role, one as the primary and one as the backup. The remaining members operate in the linecard role. In a preprovisioned Virtual Chassis or VCF, you configure the *role* for each member by associating it with the member switch's serial number, and the Virtual Chassis assigns default primary-role priorities based on

the assigned role. In a non-provisioned Virtual Chassis or VCF, you can configure roles explicitly, or the primary-role priorities of the members implicitly designate the role of each member based on the primary-role election algorithm.

The Virtual Chassis or VCF elects the member switch with the highest primary-role priority value as the primary Routing Engine. When two members have the same highest primary-role priority value among all the members, the primary-role election algorithm uses other factors to determine which of the two members functions as the primary and which is the backup. See [Understanding How the Primary in a Virtual Chassis Is Elected](#) for all the conditions considered in primary-role election.

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

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

A switch with a primary-role priority of 0 never takes the primary or backup role.

## Default

128

## Options

*number*—Primary-role priority value.

- **Range:** 0 through 255

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 9.0.

## 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](#)

## member

### IN THIS SECTION

- [Syntax | 331](#)
- [Hierarchy Level | 331](#)
- [Description | 332](#)
- [Default | 332](#)
- [Options | 332](#)
- [Required Privilege Level | 332](#)
- [Release Information | 332](#)

## 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](#)

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

## no-management-vlan

### IN THIS SECTION

- [Syntax | 333](#)
- [Hierarchy Level | 333](#)
- [Description | 333](#)
- [Required Privilege Level | 334](#)
- [Release Information | 334](#)

### Syntax

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

### Hierarchy Level

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

### Description

Remove the specified member's out-of-band management port from the virtual management Ethernet (VME) global management VLAN of the Virtual Chassis or VCF configuration.

For a member that is functioning in a linecard role, you can use this configuration to reserve the member's management Ethernet port for local troubleshooting:

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

You cannot configure the IP address for a local management Ethernet port using the CLI or the J-Web interface. To do this, you need to use the shell `ifconfig` command.

## Required Privilege Level

system—To view this statement in the configuration.

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

## Release Information

Statement introduced in Junos OS Release 9.0.

## RELATED DOCUMENTATION

[Understanding Global Management of a Virtual Chassis](#)

*Understanding Virtual Chassis Fabric Configuration*

## no-split-detection

### IN THIS SECTION

- [Syntax | 335](#)
- [Hierarchy Level | 335](#)
- [Description | 335](#)
- [Default | 335](#)
- [Required Privilege Level | 335](#)

## 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 | 213](#)

*Disabling Split and Merge in a Virtual Chassis*

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

## preprovisioned

### IN THIS SECTION

- [Syntax | 336](#)
- [Hierarchy Level | 336](#)
- [Description | 336](#)
- [Required Privilege Level | 337](#)
- [Release Information | 337](#)

### 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*

## redundancy (Graceful Switchover)

#### IN THIS SECTION

- [Syntax | 338](#)
- [Hierarchy Level | 338](#)
- [Description | 338](#)
- [Default | 338](#)
- [Required Privilege Level | 338](#)
- [Release Information | 338](#)

## Syntax

```

redundancy {
  failover {
    on-disk-failure;
    on-loss-of-keepalives;
  }
  graceful-switchover;
}

```

## Hierarchy Level

[edit [chassis \(EX Series\)](#)]

## Description

Enable redundant Routing Engines on a Virtual Chassis with two or more member switches or on a Virtual Chassis Fabric, on a standalone EX6200 or EX8200 switch with more than one Routing Engine.

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

## Default

Redundancy is enabled for the Routing Engines.

## 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 9.2.

## RELATED DOCUMENTATION

*graceful-switchover*

*Configuring Graceful Routing Engine Switchover in a Virtual Chassis*

[Configuring Graceful Routing Engine Switchover](#)

[Installing Software on an EX Series Switch with Redundant Routing Engines \(CLI Procedure\)](#)

[High Availability Features for EX Series Switches Overview](#)

## role

### IN THIS SECTION

- [Syntax | 339](#)
- [Hierarchy Level | 339](#)
- [Description | 339](#)
- [Options | 340](#)
- [Required Privilege Level | 341](#)
- [Release Information | 341](#)

### 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 | 342](#)
- [Hierarchy Level | 342](#)
- [Description | 342](#)
- [Options | 342](#)
- [Required Privilege Level | 343](#)
- [Release Information | 343](#)

### 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 | 344](#)
- [Hierarchy Level | 344](#)
- [Description | 344](#)
- [Default | 344](#)
- [Options | 344](#)
- [Required Privilege Level | 346](#)
- [Release Information | 346](#)

## 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:** *sk* to specify KB, *sm* to specify MB, or *sg* 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*

## vc-port

### IN THIS SECTION

- [Syntax | 347](#)
- [Hierarchy Level | 347](#)
- [Description | 347](#)
- [Default | 348](#)
- [Options | 348](#)
- [Required Privilege Level | 348](#)
- [Release Information | 348](#)

## Syntax

```
vc-port {  
    lag-hash (packet-based | source-port-based);  
}
```

## Hierarchy Level

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

## Description

Enable hashing of link aggregation group (LAG) network traffic over a dedicated trunk port within a Virtual Chassis, and select how the traffic within the dedicated trunk port is hashed.



**BEST PRACTICE:** Do not configure this statement unless you have a compelling reason to configure it. Configuration of this statement is optional and is only useful in a few types of network setups.

**Default**

source-port-based

**Options**

<b>lag-hash</b>	Select how to direct LAG traffic through the dedicated trunk port using either of the following options:	
	<b>packet-based</b>	Hashes all incoming LAG network traffic on the dedicated trunk port based on the packet.
	<b>source-port-based</b>	Hashes all incoming LAG network traffic on the dedicated trunk port based on the source.

**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 12.1.

**vcp-snmp-statistics**

**IN THIS SECTION**

- [Syntax | 349](#)
- [Hierarchy Level | 349](#)

- [Description | 349](#)
- [Default | 349](#)
- [Required Privilege Level | 349](#)
- [Release Information | 349](#)

## Syntax

```
vcp-snmp-statistics;
```

## Hierarchy Level

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

## Description

Enable SNMP monitoring of the Virtual Chassis ports (VCPs) for all VCPs in the Virtual Chassis or Virtual Chassis Fabric (VCF).

When this statement is enabled, SNMP gathers statistics on the Junos VCP MIBs. You can retrieve the statistics gathered by SNMP for these MIBs by using the [show snmp mib](#) command with the `walk` and `ascii` options and specifying `jnxVirtualChassisPortInPkts`.

## Default

SNMP is disabled by default on devices running Junos OS.

## 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 14.1X53-D40 and 15.1R5 only.

Statement deprecated in Junos OS Releases after 14.1X53 and 15.1.

## RELATED DOCUMENTATION

*Understanding EX Series Virtual Chassis*

*Understanding QFX Series Virtual Chassis*

*Understanding Virtual Chassis Components*

## virtual-chassis

### IN THIS SECTION

- [Syntax | 350](#)
- [Hierarchy Level | 351](#)
- [Description | 351](#)
- [Default | 352](#)
- [Required Privilege Level | 352](#)
- [Release Information | 352](#)

## 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 | 354](#)

[Operational Commands | 376](#)

---

# Routine Monitoring

## IN THIS CHAPTER

- [Command Forwarding Usage with EX Series and QFX Series Virtual Chassis | 354](#)
- [Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member | 362](#)
- [Verifying That Virtual Chassis Ports Are Operational | 364](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)
- [Verifying That Graceful Routing Engine Switchover Is Working in the Virtual Chassis | 369](#)
- [Verifying the Setting for the PIC Mode on an EX4500 Switch in a Virtual Chassis | 371](#)
- [Verifying the Setting for the Virtual Chassis Mode on EX4200, EX4500, and EX4550 Switches | 372](#)
- [Verifying Connectivity Between Virtual Chassis Member Devices | 374](#)

## Command Forwarding Usage with EX Series and QFX Series Virtual Chassis

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

You can always specify that these commands be applied to all member switches in the Virtual Chassis by using the `all-members` option, or to a specific member switch by using the `member-member-id` option. If neither option is specified, the default command forwarding behavior, which varies by command, is used. See the **Default** row in [Table 25 on page 355](#) to learn the command forwarding behavior for a specific command.

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

[Table 25 on page 355](#) provides a list of commands that can be run either on all members of the Virtual Chassis configuration or on a specific member switch.

Table 25: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration

Command	Purpose	all-members	member- <i>member-id</i>	Default
request support information	<p>Use this command when you contact JTAC about your component problem. This command is the equivalent of using the following CLI commands:</p> <ul style="list-style-type: none"> <li>• show version</li> <li>• show chassis firmware</li> <li>• show chassis hardware</li> <li>• show chassis environment</li> <li>• show interfaces extensive (for each configured interface)</li> <li>• show configuration (excluding any SECRET-DATA)</li> <li>• show system virtual-memory</li> </ul>	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members



**Table 25: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration**  
**(Continued)**

Command	Purpose	all-members	member- <i>member-id</i>	Default
request system partition hard- disk	Set up the hard disk for partitioning. After this command is issued, the hard disk is partitioned the next time the system is rebooted. When the hard disk is partitioned, the contents of / <b>altroot</b> and / <b>altconfig</b> are saved and restored. All other data on the hard disk is at risk of being lost.	Partitions the hard disk on all members of the Virtual Chassis configuration.	Partitions the hard disk on the specified member switch.	all-members
request system reboot	Reboot Junos OS for EX Series or QFX Series switches after a software upgrade and occasionally to recover from an error condition.	Reboots all members of the Virtual Chassis configuration.	Reboots the specified member switch.	all-members
request system snapshot	Back up the currently running and active file system.	Backs up the file systems on all members of the Virtual Chassis configuration.	Backs up the file system on the specified member switch.	all-members

**Table 25: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration**  
*(Continued)*

Command	Purpose	all-members	member- <i>member-id</i>	Default
request system storage cleanup	Free storage space on the switch by rotating log files and proposing a list of files for deletion. User input is required for file deletion.	Runs cleanup on all members of the Virtual Chassis configuration.	Runs cleanup on the specified member switch.	all-members
show log user	Display users who are viewing the system log.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	primary switch only
show system alarms	Display active system alarms.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system audit	Display the state and checksum values for file systems.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members

**Table 25: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration**  
**(Continued)**

Command	Purpose	all-members	member- <i>member-id</i>	Default
show system boot-messages	Display initial messages generated by the system kernel upon startup. These messages are the contents of <b>/var/run/dmesg.boot</b> .	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system buffers	Display information about the buffer pool that the Routing Engine uses for local traffic. Local traffic is the routing and management traffic that is exchanged between the Routing Engine and the Packet Forwarding Engine within the switch, as well as the routing and management traffic from IP (that is, from OSPF, BGP, SNMP, ping operations, and so on).	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members

**Table 25: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration**  
**(Continued)**

Command	Purpose	all-members	member- <i>member-id</i>	Default
show system connections	Display information about the active IP sockets on the Routing Engine. Use this command to verify which servers are active on a system and which connections are currently in progress.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system core-dumps	Display a core file generated by an internal Junos OS process.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system directory-usage	Display directory usage information.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	primary switch only
show system processes	Display information about software processes that are running on the switch and that have controlling terminals.	Displays information for all members of the Virtual Chassis configuration.		all-members

**Table 25: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration**  
**(Continued)**

Command	Purpose	all-members	member- <i>member-id</i>	Default
show system reboot	Display pending system reboots or halts.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system snapshot	Display information about the backup software that is located in the / <b>altroot</b> and / <b>altconfig</b> file systems. To back up software, use the request system snapshot command.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system software	Display the Junos OS extensions loaded on your switch.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system statistics	Display systemwide protocol-related statistics.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members

**Table 25: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration**  
**(Continued)**

Command	Purpose	all-members	member- <i>member-id</i>	Default
show system storage	Display statistics about the amount of free disk space in the switch's file systems.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system uptime	Display the current time and information about how long the switch, the switch software, and any existing protocols have been running	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
show system users	Show all users who are currently logged in.	Shows all users who are currently logged in to any members of the Virtual Chassis configuration.	Shows all users who are currently logged in to the specified member switch.	all-members
show system virtual-memory	Display the usage of Junos OS kernel memory, listed first by size of allocation and then by type of usage. Use show system virtual-memory for troubleshooting with JTAC.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members

## RELATED DOCUMENTATION

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

*Virtual Chassis Overview for Switches*

*Understanding Virtual Chassis Components*

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

### IN THIS SECTION

- [Purpose | 362](#)
- [Action | 363](#)
- [Meaning | 363](#)

### Purpose

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

The role and member ID of the member switch are displayed on the front-panel LCD (for switches that have an LCD) or in the output from the `show virtual-chassis` CLI command.

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

## Action

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

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

Virtual Chassis ID: 0000.e255.00e0

Member ID	Status	Serial No	Model	Mastership Priority	Role	Neighbor List ID, Interface
0 (FPC 0)	Prsnt	abc123	ex4200-48p	255	Master*	1 vcp-0 2 vcp-1
1 (FPC 1)	Prsnt	def456	ex4200-24t	255	Backup	2 vcp-0 0 vcp-1
2 (FPC 2)	Prsnt	abd231	ex4200-24p	128	Linecard	0 vcp-0 1 vcp-1

## Meaning

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

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

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

## RELATED DOCUMENTATION

*Configuring Primary Role of a Virtual Chassis*

*Configuring an EX4650 or a QFX Series Virtual Chassis*



## Verifying That Virtual Chassis Ports Are Operational

IN THIS SECTION

- Purpose | 364
- Action | 364
- Meaning | 366

### Purpose

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

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

### Action

Display the VCPs:

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

fpc0:

Interface or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0	Dedicated	1	Up	32000	1	vcp-0
vcp-1	Dedicated	2	Up	32000	1	vcp-1
1/0	Configured	3	Up	1000	2	vcp-255/1/0
1/1	Configured	3	Up	1000	2	vcp-255/1/1
1/2	Configured	4	Up	1000	4	vcp-255/0/20
1/3	Configured	4	Up	1000	4	vcp-255/0/21

fpc1:

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

fpc2:

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

fpc3:

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

fpc4:

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

0/21	Configured	3	Up	1000	0	vcp-255/1/3
------	------------	---	----	------	---	-------------

Meaning

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

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

RELATED DOCUMENTATION

- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)
- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) | 240](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

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

IN THIS SECTION

- [Purpose | 367](#)
- [Action | 367](#)
- [Meaning | 368](#)

## Purpose

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

You can use the J-Web monitoring functionality to view information about the switches that are members of a Virtual Chassis and their ports. J-Web monitoring is supported on EX2200, EX3300, EX4200, EX4300, EX4500, EX4550, and EX8200 switches in a Virtual Chassis. Starting with Junos OS Releases 14.1X53-D10 and 16.1R1, J-Web monitoring is supported on EX4600 switches and EX4600 switches in a Virtual Chassis.

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

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

## Action

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

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

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

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

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

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

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

## Meaning

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

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

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

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

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

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

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

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

For details about the output from CLI commands, see the **show virtual-chassis** and **show virtual-chassis vc-port statistics** command summaries.

Release History Table

Release	Description
14.1X53-D10	Starting with Junos OS Releases 14.1X53-D10 and 16.1R1, J-Web monitoring is supported on EX4600 switches and EX4600 switches in a Virtual Chassis.

RELATED DOCUMENTATION

- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)
- Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member*

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

IN THIS SECTION

- [Purpose | 369](#)
- [Action | 369](#)
- [Meaning | 371](#)

## Purpose

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

## Action

On the primary switch, verify the member ID of the backup Routing Engine:

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

Mastership	Neighbor List
------------	---------------

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

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

1. Connect to the backup Routing Engine:

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

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

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

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

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

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

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

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

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

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

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

Virtual Chassis ID: 5efa.4b7a.aae6

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

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

## Meaning

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

## RELATED DOCUMENTATION

| *Configuring Graceful Routing Engine Switchover in a Virtual Chassis*

## Verifying the Setting for the PIC Mode on an EX4500 Switch in a Virtual Chassis

### IN THIS SECTION

- Purpose | 372
- Action | 372
- Meaning | 372



## Purpose

Verify the PIC mode setting for an EX4500 or EX4550 switch in a Virtual Chassis.

## Action

To verify the current PIC mode setting:

```
user@switch> show chassis pic-mode
fpc0:
-----
Pic Mode: Intraconnect
```

## Meaning

The output shows that the PIC mode is currently set to Intraconnect. The Intraconnect mode sets the EX4500 as a standalone switch.

You must set the PIC mode to **virtual-chassis** if you want to connect the switch into a Virtual Chassis using the dedicated Virtual Chassis ports (VCPs) on the Virtual Chassis module. The default PIC mode setting is virtual chassis.

## RELATED DOCUMENTATION

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

[Resetting the EX4500 Switch to Run as a Standalone Switch \(CLI Procedure\) | 309](#)

## Verifying the Setting for the Virtual Chassis Mode on EX4200, EX4500, and EX4550 Switches

### IN THIS SECTION

● [Purpose | 373](#)

● [Action | 373](#)

## Purpose

You must configure EX4200, EX4500, and EX4550 switches into mixed Virtual Chassis mode if you want those switches to act as member switches in a mixed EX4200 and EX4500 Virtual Chassis, mixed EX4200 and EX4550 Virtual Chassis, or mixed EX4200, EX4500, and EX4550 Virtual Chassis. You must also configure a switch out of mixed Virtual Chassis mode if you remove the switch from a mixed Virtual Chassis. You must change the Virtual Chassis mode for all member switches in a Virtual Chassis if the Virtual Chassis is changed to a non-mixed Virtual Chassis..

## Action

To display the Virtual Chassis mode of any EX4200, EX4500, or EX4550 switch:

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

```
fpc0:
```

```
-----
```

```
Mixed Mode: Disabled
```

## Meaning

The output indicates that the switch is currently not in mixed Virtual Chassis mode.

## RELATED DOCUMENTATION

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

## Verifying Connectivity Between Virtual Chassis Member Devices

### IN THIS SECTION

- Purpose | 374
- Action | 374
- Meaning | 375

### Purpose

Verify that a member device is reachable through the Virtual Chassis.

### Action

1. Run a Virtual Chassis device reachability test:

```
user@switch> request virtual-chassis device-reachability test-name name source-fpc source-fpc-id destination-fpc destination-fpc-id
```

2. View the test results that appear on the screen:

```
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
```

## Meaning

The test results confirm that the connection between the member devices within the Virtual Chassis is operating properly. The **Packets Count/Sent/Received/Sendfail** output shows that five ping packets were sent and five ping packets were returned. The **Unknown/Timeout/Duplicate packets received** field output shows that zero ping packets errors occurred during the test. The **Round-trip Min/Avg/Max** output shows that all of the test packets took between 181 and 3010 microseconds to traverse the path.

## RELATED DOCUMENTATION

| [Understanding Virtual Chassis Device Reachability Testing](#) | 79

# Operational Commands

## IN THIS CHAPTER

- [clear virtual-chassis vc-port statistics | 377](#)
- [request chassis pic-mode | 379](#)
- [request session member | 382](#)
- [request system software nonstop-upgrade | 383](#)
- [request virtual-chassis device-reachability | 396](#)
- [request virtual-chassis mode | 399](#)
- [request virtual-chassis recycle | 405](#)
- [request virtual-chassis renumber | 407](#)
- [request virtual-chassis vc-port \(Dedicated VCP\) | 408](#)
- [request virtual-chassis vc-port | 411](#)
- [show chassis nonstop-upgrade | 414](#)
- [show chassis pic-mode | 417](#)
- [show system uptime | 419](#)
- [show virtual-chassis active-topology | 426](#)
- [show virtual-chassis device-topology | 433](#)
- [show virtual-chassis protocol adjacency | 441](#)
- [show virtual-chassis protocol database | 446](#)
- [show virtual-chassis protocol interface | 452](#)
- [show virtual-chassis protocol route | 455](#)
- [show virtual-chassis protocol statistics | 459](#)
- [show virtual-chassis fast-failover | 464](#)
- [show virtual-chassis login | 466](#)
- [show virtual-chassis mode | 467](#)
- [show virtual-chassis | 471](#)
- [show virtual-chassis vc-path | 480](#)
- [show virtual-chassis vc-port | 485](#)

- [show virtual-chassis vc-port statistics | 492](#)

## clear virtual-chassis vc-port statistics

### IN THIS SECTION

- [Syntax | 377](#)
- [Description | 377](#)
- [Options | 377](#)
- [Required Privilege Level | 378](#)
- [Sample Output | 378](#)
- [Release Information | 379](#)

### 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.
<b>member <i>member-id</i></b>	(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 pic-mode

### IN THIS SECTION

- [Syntax | 380](#)
- [Description | 380](#)
- [Options | 380](#)
- [Required Privilege Level | 381](#)
- [Sample Output | 381](#)
- [Release Information | 381](#)



## Syntax

```
request chassis pic-mode (intraconnect | virtual-chassis)
<all-members>
<local>
<member member-id>
```

## Description

Set the PIC mode on an EX4500 switch. The PIC mode command determines whether the module that is plugged into the EX4500 switch operates as an intraconnect module or a Virtual Chassis module.

**NOTE:** Operating an EX4500 switch without the intraconnect module or the Virtual Chassis module is not supported.

**NOTE:** If you use the EX4500 switch as a standalone switch without changing the PIC mode setting to intraconnect, you will not be able to achieve 960 Gbps data rate throughput. Some of the bandwidth is reserved for the dedicated Virtual Chassis ports (VCPs).

The PIC mode must be set to virtual-chassis if the EX4500 switch is cabled into a Virtual Chassis using the dedicated VCPs on the Virtual Chassis module. This is the setting in the default factory configuration. The intraconnect PIC mode is not supported by the default factory configuration.

If you want to use the EX4500 switch as a standalone switch, you must take several steps to reset the PIC mode as intraconnect. You cannot reset the PIC mode intraconnect simply by issuing the `request chassis pic-mode intraconnect` command. See ["Resetting the EX4500 Switch to Run as a Standalone Switch \(CLI Procedure\)" on page 309](#).

Use the ["show chassis pic-mode" on page 417](#) command to verify the current PIC mode setting.

The PIC mode setting is maintained through reboots even though it is set in operational mode.

You do not need to enter this command to use the Virtual Chassis module on an EX4550 switch.

## Options

**intraconnect**            Set the PIC mode to intraconnect.

<b>virtual-chassis</b>	Set the PIC mode to Virtual Chassis.
<b>all-members</b>	(Optional) Set the PIC mode for all members of the Virtual Chassis configuration. This is the default option.
<b>local</b>	(Optional) Set the PIC mode on the member switch where the command is issued.
<b>member <i>member-id</i></b>	(Optional) Set the PIC mode on the specified member of the Virtual Chassis configuration.

## Required Privilege Level

view

## Sample Output

### request chassis pic-mode virtual-chassis

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

### request chassis pic-mode intraconnect

```
user@switch> request chassis pic-mode intraconnect
```

```
fpc0:
```

```
-----  
virtual-chassis (mode unchanged; intraconnect mode not supported on factory-default config)
```

## Release Information

Command introduced in Junos OS Release 11.1.

## RELATED DOCUMENTATION

[Verifying the Setting for the PIC Mode on an EX4500 Switch in a Virtual Chassis | 371](#)

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

[Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 271](#)

[Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) | 273](#)

## request session member

### IN THIS SECTION

- [Syntax | 382](#)
- [Description | 382](#)
- [Options | 382](#)
- [Required Privilege Level | 382](#)
- [Release Information | 382](#)

### 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 | 383](#)
- [Description | 383](#)
- [Options | 386](#)
- [Required Privilege Level | 387](#)
- [Output Fields | 387](#)
- [Sample Output | 387](#)
- [Release Information | 395](#)

### 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.

### *set* *[package-name package-name]*

(Mixed Virtual Chassis only) Locations of the different installation packages required by the different types of member switches. These packages must be for the same Junos OS release. See this command's *package-name* option for information about how to specify the installation packages.

### *force-host*

(Optional) Force adding the host software package or bundle (and ignore warnings) on EX4650, QFX5100, or QFX5120 devices.

<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.
<b>reboot</b>	(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 <a href="#">request system reboot</a> 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 \(CLI Procedure\)](#)



## request virtual-chassis device-reachability

### IN THIS SECTION

- [Syntax | 396](#)
- [Description | 396](#)
- [Options | 397](#)
- [Required Privilege Level | 397](#)
- [Output Fields | 397](#)
- [Sample Output | 398](#)
- [Release Information | 399](#)

### 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 26 on page 398 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 26: 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/Timedout/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

[Understanding Virtual Chassis Device Reachability Testing](#) | 79

## request virtual-chassis mode

### IN THIS SECTION

- [Syntax](#) | 400
- [Description](#) | 400
- [Options](#) | 403
- [Required Privilege Level](#) | 404
- [Sample Output](#) | 404
- [Sample Output](#) | 404
- [Sample Output](#) | 404
- [Release Information](#) | 404

## Syntax

```
request virtual-chassis mode
    fabric
    mixed <ieee-clause-82>
    network-port
    <disable>
    <reboot>
    <all-members> | <local> | <member member-id>
```

## Description

Enable special device modes when operating as part of a Virtual Chassis or Virtual Chassis Fabric (VCF). Supported device modes include:

- Fabric mode—Enable the device to operate as part of a VCF. You must set fabric mode on all of the devices in a VCF.
- Mixed mode—Enable the device to operate in a mixed Virtual Chassis or VCF.
- (EX4400 switches only) Network port mode—Enable the default Virtual Chassis ports (VCPs) to operate as network ports instead of as VCPs.

You must reboot the standalone device, Virtual Chassis or VCF for this command to take effect, and the setting persists across subsequent reboot operations. You can use this command's reboot option to automatically reboot the device when setting or disabling a device mode.

**NOTE:** For devices in a mixed VCF, you can set both mixed and fabric modes in the same command and then reboot the device once for both settings to take effect.

## Mixed Mode Requirements and Best Practices

You must set mixed mode on all devices in a Virtual Chassis or a VCF when the devices have differences in how they interoperate. See [Understanding Mixed EX Series and QFX Series Virtual Chassis](#) or [Understanding Mixed Virtual Chassis Fabric](#) for details on which devices can be combined to form a mixed Virtual Chassis or VCF.

Some combinations of different types of switches form non-mixed Virtual Chassis or VCFs because the devices can run the same software images. You *don't* need to configure mixed mode if the only devices in your Virtual Chassis are the following combinations of switches:

- Only EX4500 and EX4550 switches in an EX Series Virtual Chassis.
- EX2300 multigigabit switches (EX2300-24MP and EX2300-48MP models) with any other EX2300 model switches in an EX2300 Virtual Chassis.
- EX4400 multigigabit switches (EX4400-24MP and EX4400-48MP models) with any other EX4400 model switches in an EX4400 Virtual Chassis.
- Only QFX3500 and QFX3600 switches in a QFX Series Virtual Chassis.
- QFX5110 and QFX5100 switches in a QFX5110 Virtual Chassis or QFX5110 VCF.

However, EX4300 switches that are not multigigabit models combined with EX4300 multigigabit model (EX4300-48MP) switches form a mixed EX4300 Virtual Chassis. In this case, you must set mixed mode on all member switches. In addition, when you set mixed mode on the EX4300 non-multigigabit model member switches, you must include the `ieee-clause-82` option with the mode command. This option sets a special port mode (IEEE Clause 82) on the Virtual Chassis ports (VCPs) that enables them to communicate when interconnected with VCPs on EX4300 multigigabit switches.

**NOTE:** If you remove an EX4300 member switch from a mixed EX4300 Virtual Chassis with multigigabit model members, you must disable `ieee-clause-82` port mode on the switch if you want to reconfigure it as a standalone switch or use it in any other type of mixed Virtual Chassis or any non-mixed Virtual Chassis. Otherwise, the VCPs will not connect with other members in the new Virtual Chassis. To do this, use the `request virtual-chassis mode mixed ieee-clause-82 disable` command and then reboot the switch. To add the switch to any other type of mixed Virtual Chassis, enable mixed mode again without this option and reboot the switch again for the change to take effect.

Best practices when setting `mixed` mode include:

- Don't enable mixed mode on a standalone device or for a member switch that is in a non-mixed Virtual Chassis or VCF. Mixed mode reduces the maximum scaling numbers for some features on the standalone switch, Virtual Chassis, or VCF to accommodate different types of devices that can be interconnected.
- To avoid potential traffic disruptions and configuration issues, on each device we recommend setting mixed mode for a mixed Virtual Chassis or VCF before cabling them into a mixed Virtual Chassis or VCF. You must then reboot the device, Virtual Chassis or VCF for the mode change to take effect.

**NOTE:** The same best practice applies to setting fabric mode for a VCF—set this mode on the member devices before interconnecting them.

- You can change mode settings manually after a device has been added to a Virtual Chassis or VCF if you determine the settings are not the same on all the devices.

However, if you set only some of the devices in a mixed Virtual Chassis or VCF to mixed mode, the mixed Virtual Chassis or VCF might not form. If you experience this issue, enter the `request virtual-chassis mode mixed all-members` command to force setting mixed mode on all devices in the Virtual Chassis or VCF. (Include the `fabric mode` option as well for a VCF.) You can then reboot the entire Virtual Chassis or VCF, or only the devices where the mode changed. The Virtual Chassis or VCF forms when the rebooted devices are online again.

**NOTE:** On switches running Junos OS Release 11.4 or later, if you don't set any of the switches in a mixed EX4200 and EX4500 Virtual Chassis to mixed mode, a mixed EX4200 and EX4500 Virtual Chassis forms where one of the switches assumes the primary role, and all other switches take on the linecard role. If you see this behavior, enter the `request virtual-chassis mode mixed all-members` command to set all members to mixed mode and reboot the Virtual Chassis.

### Network Port Mode (EX4400 Switches Only)

If you are not using an EX4400 switch as a member switch in a Virtual Chassis, you can use this command with the `network-port` option to enable network port mode, which converts the default VCPs on the switch into network ports. By default, the two 100-Gigabit Ethernet QSFP28 ports on the rear panel are set as two logical 50-Gbps VCP interfaces each, forming a total of four logical 50-Gbps VCPs. (The default VCPs are in PIC slot 1, so the VCP interfaces are named `vcp-255/1/0` through `vcp-255/1/3`.) If you convert these ports into network port mode, they become two 100 Gigabit Ethernet network ports, or you can channelize each of them into four 25-Gigabit Ethernet or four 10-Gigabit Ethernet network ports.

You can't use one of the rear panel 100-Gigabit Ethernet ports as a VCP and the other as a network port. When you use this command to enable network port mode, the mode affects both 100-Gbps ports together.

You can use the `show virtual-chassis mode` command to see the current port mode and what the port mode will be the next time the switch is rebooted. The two values might be different if you changed the mode but have not rebooted the switch yet.

To disable network port mode and return these ports to their default settings as VCPs, use the `network-port` and `disable` options with the `request virtual-chassis mode` command. You must reboot the switch for

network port mode changes to take effect, so you can include the reboot option in the same command. For example:

```
request virtual-chassis mode network-port disable reboot
```

You can use the `request virtual-chassis vc-port delete` command to disable one of these ports as a VCP, but that action doesn't change the port mode to network port mode. You must enter the `request virtual-chassis mode network-port` command (and reboot the switch) to use the port as a network port. Also, if you run the `request virtual-chassis vc-port delete interface interface-name` command to disable vcp-255/1/0, that command disables both logical ports 0 and 1 (vcp-255/1/0 and vcp-255/1/1). Similarly, specifying to disable vcp-255/1/2 disables both logical ports 2 and 3 (vcp-255/1/2 and vcp-255/1/3).

### Options

- all-members** (Optional) Set the Virtual Chassis mode for all members of the Virtual Chassis or VCF.
  
- disable** Disable the specified Virtual Chassis mode setting if it was previously enabled. You must include at least one of the supported modes that you want to disable with this option: `fabric`, `mixed`, or `network-port`.  
  
 The `request virtual-chassis mode mixed disable` command does not also automatically disable the mixed mode `ieee-clause-82` port mode option. To disable `ieee-clause-82` port mode on the device, you must include that option in the `disable` command as follows:  
  

```
request virtual-chassis mode mixed ieee-clause-82 disable
```
  
- fabric** Set the device into fabric mode so it can participate as a member device in a VCF.
  
- local** (Optional) Set the Virtual Chassis mode only on the member device where the command is issued.
  
- member *member-id*** (Optional) Set the Virtual Chassis mode on the specified member of the Virtual Chassis or VCF.
  
- mixed <ieee-clause-82>** Set the device into mixed mode so that the device can participate in a mixed Virtual Chassis or mixed VCF.  
  
 You must set mixed mode with the `ieee-clause-82` option on EX4300 switches that are not multigigabit models when mixing them with EX4300 multigigabit (EX4300-48MP) model switches in an EX4300 Virtual Chassis.
  
- network-port** (EX4400 switches only) Set an EX4400 switch into network port mode, which converts the default VCPs on the switch to network port mode so you can use them as network ports instead of VCPs.



**reboot** After applying the mode change specified by the other options, reboot the device automatically. You must reboot the device for any mode change to take effect.

## Required Privilege Level

system-control

## Sample Output

**request virtual-chassis mode mixed**

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

## Sample Output

**request virtual-chassis mode fabric mixed reboot**

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

## Sample Output

**request virtual-chassis mode mixed (with IEEE Clause 82 mode for EX4300 switches mixed with EX4300 multigigabit switches in a Virtual Chassis)**

```
user@switch> request virtual-chassis mode ieee-clause-82 mixed
```

## Release Information

Command introduced in Junos OS Release 11.1.

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

ieee-clause-82 mixed-mode option introduced for EX4300 switches mixed with EX4300 multigigabit switches in a Virtual Chassis in Junos OS Release 18.2R1.

network-port mode introduced in Junos OS Release 21.1R1 for EX4400 switches in a Virtual Chassis.

## RELATED DOCUMENTATION

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

*Understanding Mixed Virtual Chassis Fabric*

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#)

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

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

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

[Adding a New Switch to an Existing EX2300, EX3400, or EX4300 Virtual Chassis](#)

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

*Verifying the Virtual Chassis Fabric Mode Settings*

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

## request virtual-chassis recycle

### IN THIS SECTION

- [Syntax | 405](#)
- [Description | 406](#)
- [Options | 406](#)
- [Required Privilege Level | 406](#)
- [Sample Output | 406](#)
- [Sample Output | 406](#)
- [Release Information | 406](#)

### 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*

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

## request virtual-chassis renumber

### IN THIS SECTION

- [Syntax | 407](#)
- [Description | 407](#)
- [Options | 407](#)
- [Required Privilege Level | 408](#)
- [Sample Output | 408](#)
- [Release Information | 408](#)

### 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*

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

## request virtual-chassis vc-port (Dedicated VCP)

### IN THIS SECTION

- [Syntax | 409](#)
- [Description | 409](#)
- [Options | 409](#)
- [Additional Information | 409](#)
- [Required Privilege Level | 410](#)

- Sample Output | 410
- Release Information | 410

Syntax

```
request virtual-chassis vc-port set interface vcp-interface-name
<member member-id> <disable>
```

Description

Disable or enable a Virtual Chassis port (VCP) on a dedicated VCP.

Configure a Gigabit Ethernet link on an EX8200 switch as a VCP in an EX8200 Virtual Chassis. You can use this command only to configure a link between two EX8200 member switches as a VCP link; all other links in an EX8200 Virtual Chassis are automatically VCP links.

Options

- |  |   |
|--|---|
| <b>interface <i>vcp-interface-name</i></b> | Name of the interface to enable or disable.<br><br>If you omit <b>member <i>member-id</i></b> in an EX4200 Virtual Chassis, EX4500 Virtual Chassis, EX4550 Virtual Chassis, mixed Virtual Chassis, this command defaults to disabling or enabling the dedicated VCP on the switch where the command is issued. The dedicated VCPs are enabled in the factory default configuration. |
| <b>member <i>member-id</i></b>             | (Optional on EX4200, EX4500, and EX4550 switches; required on EX8200 switches)<br>Enable or disable the specified VCP on the specified member of the Virtual Chassis configuration.<br><br>This option must be specified when using this command to configure a VCP link between two EX8200 switches.   |
| <b>disable</b>                             | (Optional) Disable the specified VCP. If you omit this keyword, the command enables the dedicated VCP.  |

Additional Information

## Required Privilege Level

system-control

## Sample Output

### request virtual-chassis vc-port set interface vcp-0 disable (EX4200 Virtual Chassis)

```
user@switch> request virtual-chassis vc-port set interface vcp-0 disable
```

To check the results of this command, use the *show virtual-chassis vc-port* command.

### request virtual-chassis vc-port set fpc-slot 4 pic-slot 0 port 2 member 1 (EX8200 Virtual Chassis)

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

To check the results of this command, use the *show virtual-chassis vc-port* command.

### request virtual-chassis vc-port set interface vcp-0 member 3 disable (EX8200 Virtual Chassis)

```
user@external-routing-engine> request virtual-chassis vc-port set interface vcp-0 member 3
disable
```

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.

## RELATED DOCUMENTATION

*request virtual-chassis vc-port*

*show virtual-chassis vc-port*

*show virtual-chassis vc-port statistics*

*clear virtual-chassis vc-port statistics*

## request virtual-chassis vc-port

### IN THIS SECTION

- Syntax | 411
- Description | 411
- Options | 412
- Required Privilege Level | 413
- Sample Output | 413
- Release Information | 413

### 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</b> <i>interface-name</i>	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</b> <i>member-id</i>	(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](#)

## show chassis nonstop-upgrade

### IN THIS SECTION

- [Syntax | 414](#)
- [Description | 414](#)
- [Required Privilege Level | 415](#)
- [Output Fields | 415](#)
- [Sample Output | 415](#)
- [Release Information | 416](#)

### 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 27 on page 415 lists the output fields for the show chassis nonstop-upgrade command. Output fields are listed in the approximate order in which they appear.

Table 27: show chassis nonstop-upgrade Output Fields

Field Name	Field Description
Item	Line card slot number.
Status	State of line card: <ul style="list-style-type: none"><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>
Reason	Reason for the state (if the line card is offline).

Sample Output

show chassis nonstop-upgrade (EX8200 Switch)

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

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

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

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

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

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

**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 chassis pic-mode

### IN THIS SECTION

- [Syntax | 417](#)
- [Description | 417](#)
- [Options | 418](#)
- [Required Privilege Level | 418](#)
- [Sample Output | 418](#)
- [Release Information | 418](#)

## Syntax

```
show chassis pic-mode
<all-members>
<local>
<member member-id>
```

## Description

Display the PIC mode on an EX4500 switch.

The PIC mode must be set to **virtual-chassis** if the EX4500 switch is cabled into a Virtual Chassis using the dedicated Virtual Chassis ports (VCPs) on the Virtual Chassis module.

**NOTE:** This PIC mode setting is not used on an EX4550 switch.

The default PIC mode setting is **virtual-chassis**.

Use the `request chassis pic-mode` command to configure the PIC mode setting.

**Options**

<b>none</b>	Display the PIC mode on the member switch where the command is issued.
<b>all-members</b>	(Optional) Display the PIC mode for all members of the Virtual Chassis configuration.
<b>local</b>	(Optional) Display the PIC mode on the member switch where the command is issued.
<b>member <i>member-id</i></b>	(Optional) Display the PIC mode of the specified member of the Virtual Chassis configuration.

**Required Privilege Level**

view

**Sample Output**

**show chassis pic-mode**

```
user@switch> show chassis pic-mode
fpc0:
-----
Pic Mode: Intraconnect
```

**Release Information**

Command introduced in Junos OS Release 11.1.

## RELATED DOCUMENTATION

[request chassis pic-mode](#) | 379

[Resetting the EX4500 Switch to Run as a Standalone Switch \(CLI Procedure\)](#) | 309

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#) | 255

## show system uptime

### IN THIS SECTION

- [Syntax](#) | 419
- [Syntax \(EX Series Switches\)](#) | 420
- [Syntax \(QFX Series\)](#) | 420
- [Syntax \(TX Matrix Router\)](#) | 420
- [Syntax \(TX Matrix Plus Router\)](#) | 420
- [Syntax \(MX Series Router\)](#) | 420
- [Syntax \(Junos OS Evolved\)](#) | 421
- [Description](#) | 421
- [Options](#) | 421
- [Additional Information](#) | 423
- [Required Privilege Level](#) | 423
- [Output Fields](#) | 423
- [Sample Output](#) | 424
- [Release Information](#) | 426

## Syntax

```
show system uptime
```



## Syntax (EX Series Switches)

```
show system uptime  
<all-members>  
<local>  
<member member-id>
```

## Syntax (QFX Series)

```
show system uptime  
<director-group name>  
<infrastructure name>  
<interconnect-device name>  
<node-group name>
```

## Syntax (TX Matrix Router)

```
show system uptime  
<all-chassis | all-lcc | lcc number / scc>
```

## Syntax (TX Matrix Plus Router)

```
show system uptime  
<detail>  
<all-chassis | all-lcc | lcc number | sfc number>
```

## Syntax (MX Series Router)

```
show system uptime  
<all-members>  
<invoke-on>  
<local>  
<member member-id>
```

## Syntax (Junos OS Evolved)

```
show system uptime
<node node-name>
```

## Description

Display the current time and information about how long the router or switch, router or switch software, and routing protocols have been running.

**NOTE:** Time values computed from differences in timestamps can vary due to the insertion or deletion of leap-seconds between them.

The `show system uptime` command is a little different in how it displays output in Junos OS Evolved. The `show system uptime` command by itself shows system-wide uptime information. Use the `show system uptime node node-name` command to see node-specific uptime information, where *node-name* can be `re0` | `re1` | `fpc0` | `all`.

## Options

<b>none</b>	Show time since the system rebooted and processes started.
<b>all-chassis</b>	(TX Matrix routers and TX Matrix Plus routers only) (Optional) Show time since the system rebooted and processes started on all the routers in the chassis.
<b>all-lcc</b>	(TX Matrix routers and TX Matrix Plus routers only) (Optional) On a TX Matrix router, show time since the system rebooted and processes started for all T640 routers (or line-card chassis) connected to the TX Matrix router. On a TX Matrix Plus router, show time since the system rebooted and processes started for all connected T1600 or T4000 LCCs.
<b>all-members</b>	(EX4200 switches and MX Series routers only) (Optional) Show time since the system rebooted and processes started on all members of the Virtual Chassis configuration.
<b>director-group <i>name</i></b>	(QFabric systems only) (Optional) Show time since the system rebooted and processes started on the Director group.

<b>infrastructure <i>name</i></b>	(QFabric systems only) (Optional) Show time since the system rebooted and processes started on the fabric control Routing Engine and fabric manager Routing Engine.
<b>interconnect- device <i>name</i></b>	(QFabric systems only) (Optional) Show time since the system rebooted and processes started on the Interconnect device.
<b>invoke-on</b>	(MX Series routers only) (Optional) Display the time since the system rebooted and processes started on the primary Routing Engine, backup Routing Engine, or both, on a router with two Routing Engines.
<b>lcc <i>number</i></b>	<p>(TX Matrix routers and TX Matrix Plus routers only) (Optional) On a TX Matrix router, show time since the system rebooted and processes started for a specific T640 router that is connected to the TX Matrix router. On a TX Matrix Plus router, show time since the system rebooted and processes started for a specific router that is connected to the TX Matrix Plus router.</p> <p>Replace <i>number</i> with the following values depending on the LCC configuration:</p> <ul style="list-style-type: none"> <li>• 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.</li> <li>• 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.</li> <li>• 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.</li> <li>• 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.</li> </ul>
<b>local</b>	(EX4200 switches and MX Series routers only) (Optional) Show time since the system rebooted and processes started on the local Virtual Chassis member.
<b>member <i>member-id</i></b>	(EX4200 switches and MX Series routers only) (Optional) Show time since the system rebooted and processes started on the specified member of the Virtual Chassis configuration. For EX4200 switches, replace <i>member-id</i> with a value from 0 through 9. For an MX Series Virtual Chassis, replace <i>member-id</i> with a value of 0 or 1.
<b>node <i>node-name</i></b>	(Junos OS Evolved only) (Optional) Display uptime information for the specified node. When node option is not used, display uptime information for entire system. From the system's point of view, the system booted time is the time when the system was assembled and is available in <code>/var/run/system_boot_time</code> .

<b>node-group name</b>	(QFabric systems only) (Optional) Show time since the system rebooted and processes started on the Node group.
<b>scc</b>	(TX Matrix routers only) (Optional) Show time since the system rebooted and processes started for the TX Matrix router (or switch-card chassis).
<b>sfc number</b>	(TX Matrix Plus routers only) (Optional) Show time since the system rebooted and processes started for the TX Matrix Plus router. Replace <i>number</i> with 0.

### Additional Information

By default, when you issue the `show system uptime` command on the primary Routing Engine of a TX Matrix router or a TX Matrix Plus router, the command is broadcast to all the primary Routing Engines of the LCCs connected to it in the routing matrix. Likewise, if you issue the same command on the backup Routing Engine of a TX Matrix or a TX Matrix Plus router, the command is broadcast to all backup Routing Engines of the LCCs that are connected to it in the routing matrix.

### Required Privilege Level

view

### Output Fields

[Table 28 on page 423](#) describes the output fields for the `show system uptime` command. Output fields are listed in the approximate order in which they appear.

**Table 28: show system uptime Output Fields**

Field Name	Field Description
System booted	(Only for Junos OS Evolved when node option is not used) Time system was last booted. From the system's point of view, the system booted time is the time when the system was assembled and is available in <code>/var/run/system_boot_time</code> .
System-wide users	(Only for Junos OS Evolved when node option is not used) Number of system-wide users.
Current time	Current system time in UTC.
Time Source	Reference time source that the system is locked to.

**Table 28: show system uptime Output Fields (Continued)**

Field Name	Field Description
System booted	Date and time when the Routing Engine on the router or switch was last booted and how long it has been running.
Protocols started	Date and time when the routing protocols were last started and how long they have been running.
Last configured	Date and time when a configuration was last committed. Also shows the name of the user who issued the last commit command.
<i>time</i> and up	Current time, in the local time zone, and how long the router or switch has been operational.
users	Number of users logged in to the router or switch.
load averages	Load averages for the last 1 minute, 5 minutes, and 15 minutes.

## Sample Output

### show system uptime

```

user@host> show system uptime
Current time:      2017-10-13 19:45:47 UTC
Time Source:      NTP CLOCK
System booted:    2017-10-12 20:51:41 UTC (22:54:06 ago)
Protocols started: 2017-10-13 19:33:45 UTC (00:12:02 ago)
Last configured:  2017-10-13 19:33:45 UTC (00:12:02 ago) by abc
12:45PM up 22:54, 2 users, load averages: 0.07, 0.02, 0.01

```

### show system uptime node (for Junos OS Evolved)

```

user@host> show system uptime node re0

Current time: 2017-05-16 16:13:18 PDT

```

```
Node booted: 2017-05-10 15:45:29 PDT (6d 00:27 ago)
Last configured: 2017-05-10 15:31:46 PDT (6d 00:41 ago) by root
16:13:18 up 6 days, 27 min, 1 user, load averages: 2.69, 2.58, 2.57
```

### show system uptime (QFX Series)

```
user@switch> show system uptime
Current time: 2017-08-27 03:12:30 PDT
Time Source: NTP CLOCK
System booted: 2017-08-13 17:11:54 PDT (1w6d 10:00 ago)
Protocols started: 2017-08-13 17:13:56 PDT (1w6d 09:58 ago)
Last configured: 2017-08-26 05:54:00 PDT (21:18:30 ago) by user
 3:12AM up 13 days, 10:01, 3 users, load averages: 0.00, 0.00, 0.00
```

### show system uptime (Junos OS Evolved)

```
user@router> show system uptime

System booted: 2019-02-20 19:17:52 PST (02:20:33 ago)
System-wide users: 7 users
```

Starting in Junos OS Evolved 19.1R1 release, the `show system uptime` output displays only the System booted and System-wide users information. The output does not display the details such as Current time, System booted, Protocols started, and Last configured parameters. To see node-specific uptime information, use the `node` option.

### show system uptime node re0 (Junos OS Evolved)

```
user@router> show system uptime node re0
Current time: 2019-07-09 14:24:34 PDT
Time Source: NTP CLOCK
Node booted: 2019-07-03 09:59:58 PDT (6d 04:24 ago)
Protocols started: 2019-07-03 10:01:41 PDT (6d 04:22 ago)
Last configured: 2019-07-03 10:01:10 PDT (6d 04:23 ago) by root
 2:24PM up 6 days, 4:24, 1 user, load averages: 1.25, 0.51, 0.36
```

## Release Information

Command introduced before Junos OS Release 7.4.

sfc option introduced for the TX Matrix Plus router in JUNOS Release 9.6.

node option introduced in Junos OS Evolved Release 18.3R1.

## RELATED DOCUMENTATION

[10-Gigabit Ethernet LAN/WAN PIC with XFP \(T640 Router\)](#)

[Routing Matrix with a TX Matrix Plus Router Solutions Page](#)

## show virtual-chassis active-topology

### IN THIS SECTION

- [Syntax | 426](#)
- [Description | 426](#)
- [Options | 427](#)
- [Required Privilege Level | 427](#)
- [Output Fields | 427](#)
- [Sample Output | 427](#)
- [Release Information | 432](#)

## 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.
<b>member</b> <i>member-id</i>	(Optional) Display the active topology of the specified member of the Virtual Chassis or VCF.

### Required Privilege Level

view

### Output Fields

[Table 29 on page 427](#) 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 29: 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 device-topology

## IN THIS SECTION

- [Syntax | 433](#)
- [Description | 433](#)
- [Options | 433](#)
- [Required Privilege Level | 434](#)
- [Output Fields | 434](#)
- [Sample Output | 435](#)
- [Release Information | 441](#)

## Syntax

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

## Description

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

## Options

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

## Required Privilege Level

clear

## Output Fields

Table 30 on page 434 lists the output fields for the `show virtual-chassis device-topology` command. Output fields are listed in the approximate order in which they appear.

**Table 30: show virtual-chassis device-topology Output Fields**

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

## Sample Output

### show virtual-chassis device-topology

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

-----						
				Neighbor List		
Member	Device	Status	System ID	Member	Device	Interface
0	0	Prsnt	0021.59f7.d000	8	8	vcp-0/0
				1	1	vcp-4/0/1
1	1	Prsnt	0026.888d.6800	8	8	vcp-0/0





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

fpc1:

## Neighbor List

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

fpc2:

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

fpc3:

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

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

fpc4:

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

fpc5:

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

fpc6:

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

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

Release Information

Command introduced in Junos OS Release 10.4.

RELATED DOCUMENTATION

- [Understanding Virtual Chassis Port Link Aggregation](#)
- [Understanding EX8200 Virtual Chassis Topologies](#)

show virtual-chassis protocol adjacency

IN THIS SECTION

- Syntax | 442
- Description | 442
- Options | 442
- Required Privilege Level | 443
- Output Fields | 443
- Sample Output | 444

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 <b>brief</b> option is equivalent to entering the command with no options (the default). The <b>detail</b> and <b>extensive</b> options provide identical displays.
<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 31 on page 443 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 31: 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 | 447](#)
- [Description | 447](#)
- [Options | 447](#)
- [Required Privilege Level | 447](#)
- [Output Fields | 447](#)
- [Sample Output | 448](#)
- [Release Information | 451](#)

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 brief option is equivalent to entering the command with no options (the default). The detail option provides more output than the brief option. The extensive option provides all output and is most useful for customer support personnel.
<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 32 on page 448](#) 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 32: show virtual-chassis protocol database Output Fields**

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

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

## Sample Output

### show virtual-chassis protocol database

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

```
member0:
```

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

```
member1:
```

```
-----
```

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

4 LSPs

member8:

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

4 LSPs

member9:

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

4 LSPs

## show virtual-chassis protocol database detail

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

member0:

0000.4a75.9b7c.00-00 Sequence: 0x1ddbc, Checksum: 0x3111, Lifetime: 115 secs

Neighbor: 0000.73e9.9a57.00 Interface: vcp-1/0.32768 Metric: 150

Neighbor: 0021.59f7.d000.00 Interface: vcp-1/1.32768 Metric: 150

Neighbor: 0026.888d.6800.00 Interface: vcp-1/2.32768 Metric: 150

0000.73e9.9a57.00-00 Sequence: 0xf381, Checksum: 0xe065, Lifetime: 114 secs

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

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

0021.59f7.d000.00-00 Sequence: 0x168af, Checksum: 0x8b0b, Lifetime: 118 secs

```

Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

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

member1:
-----

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

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

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

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

member8:
-----

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

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

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

```

```

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

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

member9:
-----

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

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

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

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

```

## 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 | 452](#)
- [Description | 452](#)
- [Options | 452](#)
- [Required Privilege Level | 453](#)
- [Output Fields | 453](#)
- [Sample Output | 453](#)
- [Release Information | 454](#)

## 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 brief option is equivalent to entering the command with no options (the default). The detail option provides more output than the brief option.
<b>all-members</b>	(Optional) Display VCCP interface statistics for all members of the Virtual Chassis or VCF.

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

## Required Privilege Level

clear

## Output Fields

[Table 33 on page 453](#) 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 33: 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 | 455](#)
- [Description | 455](#)
- [Options | 456](#)
- [Required Privilege Level | 456](#)
- [Output Fields | 456](#)
- [Sample Output | 457](#)
- [Release Information | 459](#)

## 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 34 on page 456](#) 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 34: 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 34: 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 | 460
- Description | 460
- Options | 460
- Required Privilege Level | 460
- Output Fields | 460
- Sample Output | 462



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 35 on page 461](#) 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 35: 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 fast-failover

### IN THIS SECTION

- [Syntax | 464](#)
- [Description | 464](#)
- [Required Privilege Level | 464](#)
- [Output Fields | 465](#)
- [Sample Output | 465](#)
- [Release Information | 465](#)

## Syntax

```
show virtual-chassis fast-failover
```

## Description

Display information about the fast failover feature in a Virtual Chassis configuration.

## Required Privilege Level

view

## Output Fields

Table 36 on page 465 lists the output fields for the `show virtual-chassis fast-failover` command. Output fields are listed in the approximate order in which they appear.

**Table 36: show virtual-chassis fast-failover Output Fields**

Field Name	Field Description
<b>Fast failover on dedicated VCP ports</b>	Indicates fast failover status on dedicated VCPs.
<b>Fast failover on XE uplink VCP ports</b>	Indicates fast failover status on XFP uplink VCPs.
<b>Fast failover on GE uplink VCP ports</b>	Indicates fast failover status on SFP uplink VCPs.

## Sample Output

### `show virtual-chassis fast-failover`

```
user@switch1> show virtual-chassis fast-failover
Fast failover on dedicated VCP ports: Enabled
Fast failover on XE uplink VCP ports: Disabled
Fast failover on GE uplink VCP ports: Enabled
```

## Release Information

Command introduced in Junos OS Release 9.3.

## RELATED DOCUMENTATION

[Understanding Fast Failover in an EX Series Virtual Chassis | 57](#)

[Configuring Fast Failover in an EX Series Virtual Chassis | 297](#)

[Disabling Fast Failover in an EX Series Virtual Chassis | 298](#)

## show virtual-chassis login

### IN THIS SECTION

- [Syntax | 466](#)
- [Description | 466](#)
- [Required Privilege Level | 466](#)
- [Sample Output | 466](#)
- [Release Information | 467](#)

### Syntax

```
show virtual-chassis login
```

### Description

Supply the address of the host that logged into the Virtual Chassis or VCF, or identify the location of the member switch that redirected the current session to a different member switch.

You might need this information for tracing or troubleshooting purposes.

### Required Privilege Level

view

### Sample Output

#### show virtual-chassis login (Direct Login to the Primary Console Port)

```
user@switch> show virtual-chassis login
Current login session initiated from host 192.0.2.3
```

## show virtual-chassis login (Backup Console Session Redirected to the Primary Console Port)

```
user@switch> show virtual-chassis login
Current login session initiated from host backup
```

### Release Information

Command introduced in Junos OS Release 9.3.

### RELATED DOCUMENTATION

*request session member*

[Understanding Global Management of a Virtual Chassis](#)

## show virtual-chassis mode

### IN THIS SECTION

- [Syntax | 468](#)
- [Description | 468](#)
- [Options | 468](#)
- [Required Privilege Level | 468](#)
- [Output Fields | 468](#)
- [Sample Output | 470](#)
- [Sample Output | 470](#)
- [Sample Output | 470](#)
- [Release Information | 471](#)



# Syntax

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

# Description

Display the Virtual Chassis or Virtual Chassis Fabric (VCF) mixed mode status.

# Options

- none** Display the Virtual Chassis or VCF mixed mode status for the device where you enter the command.
- all-members** (Optional) Display the Virtual Chassis or VCF mixed mode status for all member devices in the Virtual Chassis or VCF.
- local** (Optional) Display the Virtual Chassis or VCF mixed mode status for the device where you enter the command.
- member *member-id*** (Optional) Display the Virtual Chassis or VCF mixed mode status for the specified member device.

# Required Privilege Level

view

# Output Fields

[Table 37 on page 468](#) lists the output fields for the `show virtual-chassis mode` command.

**Table 37: show virtual-chassis mode Output Fields**

Field Name	Field Description
Mixed Mode	Specifies the mixed mode status of the member switch. Mixed mode is either <b>Enabled</b> or <b>Disabled</b> .

Table 37: show virtual-chassis mode Output Fields *(Continued)*

Field Name	Field Description
<b>Current mode</b>	<p>Shows the current mixed and fabric mode settings running on the member device or devices.</p> <p>(EX4400 switches only) Shows whether the switch has network port mode enabled to use the default Virtual Chassis ports (VCPs) as network ports instead of as VCPs.</p> <p>A device reboot is required to change the device mode. The <b>Current mode</b> and <b>Future mode after reboot</b> are different when the mode has been changed but the device has not been rebooted.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> <li>• <b>Fabric with mixed devices</b>—Fabric mode and mixed mode are enabled.</li> <li>• <b>Fabric with similar devices</b>—Fabric mode is enabled and mixed mode is disabled.</li> <li>• <b>Virtual Chassis with mixed devices</b>—Fabric mode is disabled and mixed mode is enabled.</li> <li>• <b>Virtual Chassis with similar devices</b>—Fabric mode is disabled and mixed mode is disabled.</li> <li>• (EX4400 switches only) <b>network-port mode</b>—Network port mode is enabled, which means the default VCPs on the switch function as network ports rather than as VCPs.</li> </ul>
<b>Future mode after reboot</b>	<p>Shows the mixed and fabric mode settings that you have requested to be enabled on the member device or devices the next time the device is rebooted.</p> <p>A device reboot is required to change the fabric or mixed mode. The <b>Current mode</b> and <b>Future mode after reboot</b> are different only when the mode has been changed but the device has not been rebooted.</p> <p>Outputs include:</p> <ul style="list-style-type: none"> <li>• <b>Fabric with mixed devices</b>—Fabric mode and mixed mode are enabled.</li> <li>• <b>Fabric with similar devices</b>—Fabric mode is enabled and mixed mode is disabled.</li> <li>• <b>Virtual Chassis with mixed devices</b>—Fabric mode is disabled and mixed mode is enabled.</li> <li>• <b>Virtual Chassis with similar devices</b>—Fabric mode is disabled and mixed mode is disabled.</li> <li>• (EX4400 switches only) <b>network-port mode</b>—Network port mode is enabled so the default VCPs on the switch will function as network ports rather than as VCPs.</li> </ul>

## Sample Output

**show virtual-chassis mode (EX4200 in a non-mixed Virtual Chassis)**

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

```
fpc0:
```

```
-----
```

```
Mixed Mode: Disabled
```

## Sample Output

**show virtual-chassis mode (EX4400 with network port mode enabled)**

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

```
fpc0:
```

```
-----
```

```
Current mode : network-port mode
```

```
Future mode after reboot: network-port mode
```

## Sample Output

**show virtual-chassis mode (QFX5100 in a non-mixed VCF)**

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

```
fpc0:
```

```
-----
```

```
Current mode : Fabric with similar devices
```

```
Future mode after reboot : Fabric with similar devices
```

```
fpc1:
```

```
-----
```

```
Current mode : Fabric with similar devices
```

```
Future mode after reboot : Fabric with similar devices
```

```
fpc2:
```

```
-----
Current mode : Fabric with similar devices
Future mode after reboot : Fabric with similar devices
```

```
fpc3:
```

```
-----
Current mode : Fabric with similar devices
Future mode after reboot : Fabric with similar devices
```

```
fpc4:
```

```
-----
Current mode : Fabric with similar devices
Future mode after reboot : Fabric with similar devices
```

## Release Information

Command introduced in Junos OS Release 11.1.

**Current mode** and **Future mode after reboot** fields introduced in Junos OS Release 13.2X51-D20.

## RELATED DOCUMENTATION

*request virtual-chassis mode*

*Verifying the Virtual Chassis Fabric Mode Settings*

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)](#)

## show virtual-chassis

### IN THIS SECTION

- [Syntax | 472](#)
- [Description | 472](#)
- [Options | 472](#)
- [Required Privilege Level | 472](#)

- [Output Fields | 472](#)
- [Sample Output | 475](#)
- [Release Information | 480](#)

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 38 on page 472](#) lists the output fields for the `show virtual-chassis` command. Output fields are listed in the approximate order in which they appear.

Table 38: show virtual-chassis Output Fields

Field Name	Field Description
Fabric ID	(VCF only) Assigned ID used to identify the VCF.
Fabric Mode	(VCF only) Mode of the VCF: Enabled, Disabled, or Mixed.

Table 38: show virtual-chassis Output Fields (*Continued*)

Field Name	Field Description
<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.
<b>Virtual Chassis Mode</b>	<p>Mode of the Virtual Chassis or VCF. This field indicates support for the Virtual Chassis feature and, if a Virtual Chassis is configured, if it is a mixed or homogenous Virtual Chassis. Values can be:</p> <ul style="list-style-type: none"> <li>• <b>Enabled</b>—The platform supports the Virtual Chassis feature. If a Virtual Chassis is currently configured, this is a homogenous Virtual Chassis (all members are the same type of switch).</li> <li>• <b>Disabled</b>—The switch does not support the Virtual Chassis feature.</li> </ul> <p><b>NOTE:</b> Switches that support the Virtual Chassis feature do not display this value. Even if a Virtual Chassis is not currently configured, those switches display Enabled in this field.</p> <ul style="list-style-type: none"> <li>• <b>Mixed</b>—The platform supports the Virtual Chassis feature, and is configured as a mixed mode Virtual Chassis (members consist of more than one type of switch).</li> </ul>
<b>Member ID</b>	<p>Assigned member ID and FPC:</p> <ul style="list-style-type: none"> <li>• On all EX Series Virtual Chassis except EX8200 Virtual Chassis, and on a VCF, the FPC number refers to the member ID assigned to the switch.</li> <li>• On EX8200 Virtual Chassis, member IDs are numbered 0 through 9. The FPC number indicates the slot number of the line card within the Virtual Chassis. The FPC number on member 0 is always 0 through 15. The FPC number on member 1 is always 16 through 31. The FPC number on member 2 is always 32 through 47; and so on for the members.</li> </ul>

Table 38: show virtual-chassis Output Fields (*Continued*)

Field Name	Field Description
<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>
<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</b> or <b>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>

**Table 38: show virtual-chassis Output Fields (Continued)**

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

## 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 prio	Role	Mixed Mode	Route Mode	Neighbor List 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 priority	Role	Mixed Mode	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	...	ex4200-24t	249	Master*	N	8	vcp-0
							1	vcp-1
1 (FPC 1)	Prsnt	...	ex4200-24t	248	Backup	N	0	vcp-0
							2	vcp-1
2 (FPC 2)	Prsnt	...	ex4200-48p	247	Linecard	N	1	vcp-0
							3	vcp-1
3 (FPC 3)	Prsnt	...	ex4200-24t	246	Linecard	N	2	vcp-0
							4	vcp-1
4 (FPC 4)	Prsnt	...	ex4200-48p	245	Linecard	N	3	vcp-0
							5	vcp-1

```

5 (FPC 5) Prsnt ... ex4200-48t 244 Linecard N 4 vcp-0
6 (FPC 6) Prsnt ... ex4200-48t 243 Linecard N 5 vcp-0
7 (FPC 7) Prsnt ... ex4200-24f 242 Linecard N 6 vcp-0
8 (FPC 8) Prsnt ... ex4200-24f 241 Linecard N 7 vcp-0
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: cdcd.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.cdcd
```

```
Virtual Chassis Mode: Enabled
```

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

## IN THIS SECTION

- [Syntax | 480](#)
- [Description | 481](#)
- [Options | 481](#)
- [Required Privilege Level | 481](#)
- [Output Fields | 481](#)
- [Sample Output | 483](#)
- [Release Information | 485](#)

## Syntax

```
show virtual-chassis vc-path source-interface interface-name destination-interface interface-name
```

Description

Show the forwarding path a packet takes when going from a source interface to a destination interface in a Virtual Chassis or VCF configuration.

Starting in Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, when used in VCF configurations, this command displays additional information about next hops, including when the forwarding path has multiple possible next hops within the VCF.

Options

<b>source-interface</b> <i>interface-name</i>	Name of the interface from which the packet originates in the Virtual Chassis or VCF
<b>destination-interface</b> <i>interface-name</i>	Name of the interface to which the packet is being delivered in the Virtual Chassis or VCF

Required Privilege Level

view

Output Fields

[Table 39 on page 482](#) lists the output fields for the `show virtual-chassis vc-path` command. Output fields are listed in the approximate order in which they appear.

Some output field names and display order differ between the output for a Virtual Chassis and for a VCF. When this command is used to display forwarding paths in a VCF, additional fields are included in the output to show details of multiple possible next hops. The differences are described in [Table 39 on page 482](#) and shown in sample output for each mode.

**Table 39: show virtual-chassis vc-path Output Fields**

Field Name	Field Description
Hop	<p>The hop number along the path between the source and destination interfaces. The first hop entry (Hop 0) is the packet's source, intermediate hop information represents transitions through the members within the Virtual Chassis or VCF, and the last hop entry represents arrival at the packet's destination.</p> <p>For VCF output, each Hop entry also shows information about multiple possible next hops towards the destination. See Next-hop PFE, Interface, and Bandwidth output field descriptions for details.</p>
Member (Virtual Chassis) Member-ID (VCF)	The Virtual Chassis or VCF member ID of the switch that contains the Packet Forwarding Engine for each hop through which the packet passes.
PFE-Device(Virtual Chassis) PFE(VCF)	<p>The number of the Packet Forwarding Engine in each Virtual Chassis or VCF member through which a packet passes.</p> <p>For Virtual Chassis output, the Packet Forwarding Engine in each row is the next hop of the preceding Packet Forwarding Engine, including intermediate transitions through members within the Virtual Chassis.</p> <p>VCF output is expanded on multiple rows to show more information about multiple possible next hops—each hop entry is followed by one or more Next-hop PFE output fields with interface name and bandwidth information for each possible next hop.</p>
Next-hop PFE (VCF only)	One or more possible next-hop Packet Forwarding Engine numbers for VCF member Member-ID. Each Next-hop entry includes the next-hops's interface name and bandwidth.
Interface	<p>The name of the interface through which the Packet Forwarding Engines are connected. The interface for the first hop (Hop 0) is always the source interface.</p> <p>VCF output provides more information about multiple possible next hops for each hop entry, listing the Interface and the interface Bandwidth for each possible Next-hop PFE.</p>
Bandwidth (VCF only)	The bandwidth (in Gbps) of the next-hop interface for the associated Next-hop PFE entry.

				Mstr	Mixed Route		Neighbor List		
Member ID	Status	Serial No	Model	prio	Role	Mode	Mode	ID	Interface
1 (FPC 1)	Prsnt	...	qfx5100-24q-aa	129	Backup	N	F	7	vcp-255/0/7
								8	vcp-255/0/8
								12	vcp-255/0/12
								3	vcp-255/0/3
								4	vcp-255/0/4
								5	vcp-255/0/5
								6	vcp-255/0/6
								10	vcp-255/0/10
								11	vcp-255/0/11
2 (FPC 2)	Prsnt	...	qfx5100-24q-aa	129	Master*	N	F	7	vcp-255/0/7
								8	vcp-255/0/8
								12	vcp-255/0/12
								3	vcp-255/0/3



								4	vcp-255/0/4
								5	vcp-255/0/5
								6	vcp-255/0/6
								10	vcp-255/0/10
								11	vcp-255/0/11
								9	vcp-255/0/9
3 (FPC 3)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
4 (FPC 4)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
5 (FPC 5)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
6 (FPC 6)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
7 (FPC 7)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
8 (FPC 8)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
9 (FPC 9)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
10 (FPC 10)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
11 (FPC 11)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
12 (FPC 12)	Prsnt	...	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2

user@switch> **show virtual-chassis vc-path source-interface xe-12/0/25 destination-interface xe-5/0/25**

Fabric forwarding path from xe-12/0/25 (PFE 12) to xe-5/0/25 (PFE 5)

```

Hop  0  Member-ID 12 PFE 12
  Next-hop PFE 1
    Interface vcp-255/0/1.32768 Bandwidth 40
  Next-hop PFE 2
    Interface vcp-255/0/2.32768 Bandwidth 40
Hop  1  Member-ID 1 PFE 1
  Next-hop PFE 5
    Interface vcp-255/0/5.32768 Bandwidth 40
Hop  1  Member-ID 2 PFE 2
  Next-hop PFE 5
    Interface vcp-255/0/5.32768 Bandwidth 40
Hop  2  Member-ID 5 PFE 5

```

```

user@switch> show virtual-chassis vc-path source-interface xe-12/0/25 destination-interface
xe-1/0/25
Fabric forwarding path from xe-12/0/25 (PFE 12) to xe-1/0/25 (PFE 1)
Hop  0  Member-ID  12  PFE   12
      Next-hop PFE   1
      Interface vcp-255/0/1.32768  Bandwidth   40
Hop  1  Member-ID   1  PFE    1

```

### Release Information

Command introduced in Junos OS Release 9.6.

### RELATED DOCUMENTATION

- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis](#)
- [Understanding EX Series Virtual Chassis](#)
- [Understanding QFX Series Virtual Chassis](#)
- [EX8200 Virtual Chassis Overview](#)

## show virtual-chassis vc-port

### IN THIS SECTION

- [Syntax | 486](#)
- [Description | 486](#)
- [Options | 486](#)
- [Required Privilege Level | 486](#)
- [Output Fields | 486](#)
- [Sample Output | 488](#)
- [Release Information | 492](#)

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</b> <i>member-id</i>	(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 40 on page 487](#) 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 40: 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 vcp-0 and vcp-1. The dedicated VCPs in an EX4550 Virtual Chassis are VCP-1/0, VCP-1/1, VCP-2/0, and VCP-2/1.</li> <li>Optical ports set as VCPs are named 1/0 and 1/1, 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 vcp-0.</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, vcp-1/0.</li> <li>The VCPs on EX8200 member switches are named using the <i>vcp-slot-number/pic-number/interface-number</i> convention; for instance, vcp-3/0/2.</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, vcp-255/1/0 indicates that the dedicated VCP named vcp-1/0 is part of a LAG bundle, and vcp-255/1/0 represents an uplink port previously named xe-0/1/0 that is now part of a VCP LAG bundle.</li> </ul>
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>

**Table 40: show virtual-chassis vc-port Output Fields (Continued)**

Field Name	Field Description
Trunk ID	<p>A positive-number ID assigned to a link aggregation group (LAG) formed by the Virtual Chassis. The trunk ID value is -1 if no trunk is formed. A LAG between uplink VCPs requires that the link speed be the same on connected interfaces and that at least two VCPs on one member be connected to at least two VCPs on the other member in an EX4200 or EX4500 Virtual Chassis.</p> <p>Dedicated VCP LAGs are assigned trunk IDs 1 and 2. Trunk IDs for LAGs formed with uplink VCPs therefore have values of 3 or greater.</p> <p>The trunk ID value changes if the link-adjacency state between LAG members changes; trunk membership is then allocated or deallocated.</p>
Status	<p>Interface status:</p> <ul style="list-style-type: none"> <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		Absent			
4/0/5	Configured		Absent			
4/0/4	Configured		Absent			

member8:

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

vcp-0/0	Dedicated	-1	Down	1000		
vcp-1/0	Dedicated	-1	Up	1000	9	vcp-1/0
vcp-1/1	Dedicated	-1	Up	1000	0	vcp-0/0
vcp-1/2	Dedicated	-1	Up	1000	1	vcp-0/0
vcp-1/3	Dedicated	-1	Up	1000	9	vcp-1/3
vcp-2/0	Dedicated	-1	Up	1000	0	vcp-0/1
vcp-2/1	Dedicated	-1	Up	1000	9	vcp-1/2
vcp-2/2	Dedicated	-1	Down	1000		
vcp-2/3	Dedicated	-1	Down	1000		

member9:

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Disabled	1000		
vcp-1/0	Dedicated	-1	Up	1000	8	vcp-1/0
vcp-1/1	Dedicated	-1	Down	1000		
vcp-1/2	Dedicated	-1	Up	1000	8	vcp-2/1
vcp-1/3	Dedicated	-1	Up	1000	8	vcp-1/3

### show virtual-chassis vc-port all-members

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

fpc0:

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

fpc1:

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

```

vcp-0    Dedicated      1   Up   32000    0   vcp-1
vcp-1    Dedicated      2   Up   32000    0   vcp-0
1/0      Auto-Configured -1   Up   1000     3   vcp-255/1/0

```

fpc2:

```

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

```

fpc3:

```

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

```

### show virtual-chassis vc-port (QFX5120-32C Virtual Chassis)

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

localre:

-

```

Interface  Type           Trunk  Status  Speed  Neighbor
or         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 or PIC / Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
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 | 493
- Description | 493
- Options | 493
- Required Privilege Level | 493
- Output Fields | 493
- Sample Output | 498
- Release Information | 502

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.
<b>local</b>	(Optional) Display traffic statistics for VCPs on the switch or external Routing Engine where you enter this command.
<b>member <i>member-id</i></b>	(Optional) Display traffic statistics for VCPs on the specified member of a Virtual Chassis or VCF.

Required Privilege Level

view

Output Fields

[Table 41 on page 494](#) 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 41: 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
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

**Table 41: show virtual-chassis vc-port statistics Output Fields (Continued)**

Field Name	Field Description	Level of Output
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
Oversize packets	Number of packets received on the VCP that were longer than 1518 octets (excluding framing bits, but including FCS octets) but were otherwise well formed.	detail, extensive
Undersize packets	Number of packets received on the VCP that were shorter than 64 octets (excluding framing bits but including FCS octets) and were otherwise well formed..	detail, extensive

Table 41: show virtual-chassis vc-port statistics Output Fields (*Continued*)

Field Name	Field Description	Level of Output
Jabber packets	<p>Number of packets received on the VCP that were longer than 1518 octets—excluding framing bits, but including FCS octets—and that had either an FCS error or an alignment error.</p> <p><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 runs (which are normal occurrences caused by collisions) and noise hits are counted.</p>	detail, extensive
Ifout errors	Number of outbound packets received on the VCP that could not be transmitted because of errors.	detail, extensive
Packet drop events	Number of outbound packets received on the VCP that were dropped, rather than being encapsulated and sent out of the switch as fragments. The packet drop counter is incremented if a temporary shortage of packet memory causes packet fragmentation to fail.	detail, extensive
64 octets frames	Number of packets received on the VCP (including invalid packets) that were 64 octets in length (excluding framing bits, but including FCS octets).	detail, extensive

**Table 41: show virtual-chassis vc-port statistics Output Fields (Continued)**

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

# Sample Output

## show virtual-chassis vc-port statistics

```

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

{master:0}

```

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

```

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

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

member8:
-----

```

```
member9:
-----
```

**show virtual-chassis vc-port statistics brief**

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

{master:0}
```

**show virtual-chassis vc-port statistics extensive**

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

                RX                TX

Port: internal-0/24
Total octets:    0                0
Total packets:   0                0
Unicast packets: 0                0
Broadcast packets: 0              0
Multicast packets: 0              0
MAC control frames: 0              0
CRC alignment errors: 0
Oversize packets: 0
Undersize packets: 0
```



```

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

```

...

```

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

```

```

Port: vcp-1
Total octets:      0      0
Total packets:     0      0
Unicast packets:   0      0
Broadcast packets: 0      0

```

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

...

{master:0}
```

**show virtual-chassis vc-port statistics member 0**

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

{master:0}
```

## 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 | 504

---

## CHAPTER 7

# Troubleshooting Procedures

**IN THIS CHAPTER**

- [Troubleshooting an EX Series Virtual Chassis | 504](#)

## Troubleshooting an EX Series Virtual Chassis

**IN THIS SECTION**

- [A Disconnected Member Switch's ID Is Not Available for Reassignment | 504](#)
- [Load Factory Default Does Not Commit on a Multimember Virtual Chassis | 505](#)
- [The Member ID Persists When a Member Switch Is Disconnected From a Virtual Chassis | 506](#)
- [A Member Switch Is Not Participating in a Mixed Virtual Chassis | 507](#)
- [Unknown Traffic Looping Occurs After Configuring an Uplink Port as a Redundant VCP with a Dedicated VCP | 509](#)

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

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

**IN THIS SECTION**

- [Problem | 505](#)
- [Solution | 505](#)

## Problem

## Description

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

## Solution

When you disconnect a member of a Virtual Chassis configuration, the primary retains the member ID and member configuration in its configuration database. Output from the `show virtual-chassis` command continues to display the member ID of the disconnected member with a status of `NotPrsnt`.

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

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

### IN THIS SECTION

● [Problem | 505](#)

● [Solution | 505](#)

## Problem

## Description

The `load factory-default` command fails on a multimember Virtual Chassis.

## Solution

The `load factory-default` command is not supported on a multimember Virtual Chassis configuration. For information on how to revert the switches in the Virtual Chassis to factory default settings, see *Reverting to the Default Factory Configuration for the EX Series Switch*.

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

### IN THIS SECTION

● Problem | 506

● Solution | 506

### Problem

### Description

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

### Solution

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

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

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

1. To change the member ID to 0:

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

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

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

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

### IN THIS SECTION

● Problem | 507

● Solution | 507

### Problem

### Description

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

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

### Solution

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

To verify the Virtual Chassis mode:

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

```
fpc0:
```

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

```
fpc1:
```

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

```
fpc2:
```

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

```
fpc3:
```

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

```
fpc4:
```



```
-----  
Mixed Mode: Disabled
```

```
fpc5:
```

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

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

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

(EX4500 switch only) To verify the PIC mode:

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

```
fpc0:
```

```
-----  
Pic Mode: Not-Applicable
```

```
fpc1:
```

```
-----  
Pic Mode: Not-Applicable
```

```
fpc2:
```

```
-----  
Pic Mode: Not-Applicable
```

```
fpc3:
```

```
-----  
Pic Mode: Not-Applicable
```

```
fpc4:
```

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

```
fpc5:
```

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

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

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

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

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

## Unknown Traffic Looping Occurs After Configuring an Uplink Port as a Redundant VCP with a Dedicated VCP

### IN THIS SECTION

- [Problem | 509](#)
- [Solution | 509](#)

### Problem

### Description

In a Virtual Chassis comprised of EX4200, EX4500, or EX4550 switches, you observe unrecoverable looping of unknown unicast or multicast traffic following the addition of a redundant VCP link between two member switches, when the two members are connected by a dedicated VCP link and the redundant link was created by converting uplink ports to VCPs.

This behavior can occur whether the redundant VCP link is created by setting the ports manually as VCPs or if the automatic VCP conversion feature is invoked and converts the ports into VCPs automatically.

### Solution

Reboot the Virtual Chassis to properly detect the converted VCP as a redundant link with the dedicated VCP link.

After the conversion from a network port to a VCP, the egress filter table is not updated and the redundant VCP remains enabled for forwarding, which causes the looping behavior. The reboot process detects the converted port as a VCP and brings it up as disabled for forwarding.

As a result, we do not recommend connecting redundant converted uplink VCP ports between members already connected by dedicated VCPs on an active Virtual Chassis; instead, plan to add redundant uplink VCP connections during a maintenance window that can include a Virtual Chassis reboot cycle. This recommendation also applies when adding a new member to an existing active Virtual Chassis where

you are adding redundant VCP links between the new member and one of its neighbors that mix dedicated VCPs and converted uplink VCPs.

## RELATED DOCUMENTATION

---

[Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis | 366](#)

---

[Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) | 245](#)

---

[Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) | 255](#)

---

[Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) | 250](#)