



Junos[®] OS for EX Series Ethernet Switches

EX2200, EX3300, EX4200, EX4500 and EX4550
Virtual Chassis

Release

14.1X53



Modified: 2017-01-03

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

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

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

Junos[®] OS for EX Series Ethernet Switches EX2200, EX3300, EX4200, EX4500 and EX4550 Virtual Chassis
Release 14.1X53
Copyright © 2016, Juniper Networks, Inc.
All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	xv
	Documentation and Release Notes	xv
	Supported Platforms	xv
	Using the Examples in This Manual	xv
	Merging a Full Example	xvi
	Merging a Snippet	xvi
	Documentation Conventions	xvii
	Documentation Feedback	xix
	Requesting Technical Support	xix
	Self-Help Online Tools and Resources	xix
	Opening a Case with JTAC	xx
Part 1	Overview	
Chapter 1	Virtual Chassis Overview	3
	EX Series Virtual Chassis Overview	3
	Basic Configuration of a Virtual Chassis	5
	Expanding Configurations—Within a Single Wiring Closet and Across Wiring Closets	6
	Global Management of Member Switches in a Virtual Chassis	7
	High Availability Through Redundant Routing Engines	7
	Adaptability as an Access Switch or Distribution Switch	7
	Understanding EX2200 Virtual Chassis	8
	Understanding EX Series Virtual Chassis Components	9
	Maximum Number of Switches per Virtual Chassis	10
	Virtual Chassis Ports (VCPs)	11
	Master Role	12
	Backup Role	13
	Linecard Role	13
	Member Switch and Member ID	14
	Mastership Priority	14
	Mixed Virtual Chassis	15
	Virtual Chassis Identifier (VCID)	15
	Understanding Mixed EX Series and QFX Series Virtual Chassis or Virtual Chassis Fabric	16
	Virtual Chassis Fabric Summary	16
	Understanding Mixed Virtual Chassis Fabric	17
	Virtual Chassis Summary for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 Switches	17

Understanding the Routing Engine Role in a Mixed Virtual Chassis Using EX4300, EX4600, QFX3500, QFX3600, or QFX5100 Member Switches	18
Understanding EX4300, QFX3500, QFX3600, and QFX5100 Switches in a Virtual Chassis	19
Understanding Mixed EX4300 and EX4600 Virtual Chassis	19
Understanding EX4200, EX4500, and EX4550 Switches in a Mixed Virtual Chassis	19
Understanding How the Master in a Virtual Chassis Is Elected	20
Understanding Software Upgrade in an EX Series Virtual Chassis	21
Understanding Global Management of a Virtual Chassis	22
Understanding Nonvolatile Storage in a Virtual Chassis	25
Nonvolatile Memory Features	25
Understanding the High-Speed Interconnection of the Dedicated Virtual Chassis Ports Connecting EX4200, EX4500, and EX4550 Member Switches	25
Understanding EX Series Virtual Chassis Port Link Aggregation	26
Understanding EX Series Virtual Chassis Configuration	28
Understanding EX Series Virtual Chassis Switch Version Compatibility	30
Understanding Fast Failover in an EX Series Virtual Chassis	30
Supported Topologies for Fast Failover	31
How Fast Failover Works	31
Fast Failover in a Ring Topology Using Dedicated VCPs	31
Fast Failover in a Ring Topology Using Uplink Module VCPs	33
Fast Failover in a Virtual Chassis Configuration Using Multiple Ring Topologies	35
Effects of Topology Changes on a Fast Failover Configuration	36
Understanding Split and Merge in a Virtual Chassis	37
What Happens When a Virtual Chassis Configuration Splits	37
Merging Virtual Chassis Configurations	38
Understanding Automatic Software Update on Virtual Chassis Member Switches	40
Automatic Software Update Basics	40
Automatic Software Update Restrictions	40
Understanding Virtual Chassis Device Reachability Testing	42
Understanding MAC Address Assignment on a Virtual Chassis	43
Understanding High Availability on an EX Series Virtual Chassis	44

Part 2

Chapter 2

Configuration

Configuration Examples 49

Example: Configuring an EX2200 Virtual Chassis with Four Member Switches	50
Example: Configuring an EX3300 Virtual Chassis with a Master and Backup ...	54
Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet	58
Example: Configuring an EX4500 Virtual Chassis with a Master and Backup in a Single Wiring Closet	63
Example: Expanding an EX3300 Virtual Chassis	67
Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet	71

Example: Adding EX4500 Switches to a Preprovisioned Virtual Chassis	77
Example: Adding an EX4500 Switch to a Nonprovisioned Virtual Chassis	81
Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration	85
Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets	91
Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets	98
Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File	104
Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis	116
Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch	120
Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch	126
Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails	131
Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge	134
Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports	136
Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches	144
Chapter 3 Configuration Tasks	147
Configuring an EX2200 Virtual Chassis (CLI Procedure)	148
Configuring an EX2200 Virtual Chassis with a Preprovisioned Configuration File	149
Configuring an EX2200 Virtual Chassis with a Nonprovisioned Configuration File	150
Configuring an EX3300 Virtual Chassis (CLI Procedure)	152
Configuring an EX3300 Virtual Chassis with a Preprovisioned Configuration File	153
Configuring an EX3300 Virtual Chassis with a Nonprovisioned Configuration File	154
Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) . .	156
Configuring an EX4200, EX4500, or EX4550 Virtual Chassis with a Preprovisioned Configuration File	156
Configuring an EX4200, an EX4500, or EX4550 Virtual Chassis with a Nonprovisioned Configuration File	158

Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure)	160
Configuring an EX2200, EX2200-C, EX3300, EX4200, EX4300, EX4500, EX4550, or EX4600 Virtual Chassis (J-Web Procedure)	160
Enabling Virtual Chassis Mode on an EX8200 Switch (J-Web Procedure)	162
Configuring an EX8200 Virtual Chassis (J-Web Procedure)	162
Preprovision the Virtual Chassis	163
Configure Virtual Chassis Members	163
Configure Virtual Chassis Ports	163
Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure)	165
Installing Software on a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Switches (CLI Procedure)	168
Adding a New Switch to an Existing EX3300 Virtual Chassis (CLI Procedure) . .	169
Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis (CLI Procedure)	171
Adding a New Switch to an Existing Virtual Chassis Within the Same Wiring Closet	171
Adding a New Switch from a Different Wiring Closet to an Existing Virtual Chassis	173
Adding a New Switch to an Existing Preprovisioned Virtual Chassis Using Autoprovisioning	175
Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis (CLI Procedure)	176
Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis (CLI Procedure)	178
Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis (CLI Procedure)	180
Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure)	181
Remove, Repair, and Reinstall the Same Switch	182
Remove a Member Switch, Replace It with a Different Switch, and Reapply the Old Configuration	182
Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch	184
Removing an EX4200, EX4500, or EX4550 Switch From a Mixed Virtual Chassis (CLI Procedure)	184
Configuring Mastership of a Virtual Chassis (CLI Procedure)	185
Configuring Mastership Using a Preprovisioned Configuration File	186
Configuring Mastership Using a Configuration File That Is Not Preprovisioned	186
Setting a Port on an EX2200 Switch as a Virtual Chassis Port (CLI Procedure)	187
Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port (CLI Procedure)	188
Setting an Uplink VCP Between the Member Switches	190
Setting an Uplink VCP on a Standalone Switch	191

	Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch (CLI Procedure)	193
	Setting an Uplink Port as a Virtual Chassis Port Using the LCD Panel	194
	Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis (CLI Procedure)	195
	Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis (CLI Procedure)	196
	Configuring Fast Failover in an EX Series Virtual Chassis	197
	Disabling Fast Failover in an EX Series Virtual Chassis	198
	Disabling Split and Merge in a Virtual Chassis (CLI Procedure)	199
	Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure)	199
	Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure)	201
	Configuring Graceful Routing Engine Switchover in a Virtual Chassis (CLI Procedure)	201
	Resetting the EX4500 Switch to Run as a Standalone Switch (CLI Procedure)	202
	Enabling VLAN Pruning for Broadcast, Multicast, and Unknown Unicast Traffic in an EX Series Virtual Chassis (CLI Procedure)	203
Chapter 4	Configuration Statements	205
	[edit virtual-chassis] Configuration Statement Hierarchy	205
	Supported Statements in the [edit virtual-chassis] Hierarchy Level	206
	Unsupported Statements in the [edit virtual-chassis] Hierarchy Level	207
	auto-sw-update	208
	fast-failover	210
	graceful-restart (Enabling Globally)	211
	graceful-switchover	212
	id	213
	lag-hash	214
	location (Virtual Chassis)	215
	mac-persistence-timer	216
	mastership-priority	217
	member	219
	no-management-vlan	220
	no-split-detection	221
	package-name	222
	preprovisioned	223
	redundancy (Graceful Switchover)	224
	role	225
	serial-number	228
	traceoptions (Virtual Chassis)	229
	vc-port	232
	vcp-snmp-statistics	233
	virtual-chassis	234

Part 3**Administration****Chapter 5****Routine Monitoring 239**

Command Forwarding Usage with an EX Series Virtual Chassis 239

Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual
Chassis Member 244

Verifying That Virtual Chassis Ports Are Operational 245

Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual
Chassis 247Verifying That Graceful Routing Engine Switchover Is Working in the Virtual
Chassis 249Verifying the Setting for the PIC Mode on an EX4500 Switch in a Virtual
Chassis 250Verifying the Setting for the Virtual Chassis Mode on EX4200, EX4500, and
EX4550 Switches 250

Verifying Connectivity Between Virtual Chassis Member Devices 251

Chapter 6**Operational Commands 253**

clear virtual-chassis vc-port statistics 254

request chassis pic-mode 256

request session member 258

request virtual-chassis device-reachability 259

request virtual-chassis mode 261

request virtual-chassis recycle 264

request virtual-chassis renumber 265

request virtual-chassis vc-port 266

request virtual-chassis vc-port 268

show chassis pic-mode 270

show snmp mib 272

show system uptime 275

show virtual-chassis active-topology 280

show virtual-chassis device-topology 285

show virtual-chassis protocol adjacency 291

show virtual-chassis protocol database 295

show virtual-chassis protocol interface 299

show virtual-chassis protocol route 302

show virtual-chassis protocol statistics 305

show virtual-chassis fast-failover 308

show virtual-chassis login 309

show virtual-chassis mode 310

show virtual-chassis 313

show virtual-chassis vc-path 318

show virtual-chassis vc-port 322

show virtual-chassis vc-port statistics 326

Part 4	Troubleshooting	
Chapter 7	Troubleshooting Procedures	335
	Troubleshooting an EX Series Virtual Chassis	335
	A Disconnected Member Switch's ID Is Not Available for Reassignment . . .	335
	Load Factory Default Does Not Commit on a Multimember Virtual Chassis	335
	The Member ID Persists When a Member Switch Is Disconnected From a Virtual Chassis	336
	A Member Switch Is Not Participating in a Mixed Virtual Chassis	336
	Unknown Traffic Looping Occurs After Configuring an Uplink Port as a Redundant VCP with a Dedicated VCP	337

List of Figures

Part 1	Overview	
Chapter 1	Virtual Chassis Overview	3
	Figure 1: Console Session Redirection (EX4200 Virtual Chassis Pictured)	23
	Figure 2: Management Ethernet Port Redirection to the VME Interface	24
	Figure 3: Normal Traffic Flow in a Ring Topology Using Dedicated VCPs	32
	Figure 4: Traffic Redirected by Fast Failover After a Dedicated VCP Link Failure	33
	Figure 5: Normal Traffic Flow in a Ring Topology Using SFP Uplink VCPs	34
	Figure 6: Traffic Redirected by Fast Failover After SFP Uplink VCP Link Failure	35
	Figure 7: Traffic Redirected by Fast Failover After VCP Link Failures in a Topology with Multiple Rings	36
Part 2	Configuration	
Chapter 2	Configuration Examples	49
	Figure 8: Basic EX4200 Virtual Chassis with Master and Backup	60
	Figure 9: Basic EX4500 Virtual Chassis with Master and Backup	64
	Figure 10: Expanded EX4200 Virtual Chassis in a Single Wiring Closet	73
	Figure 11: Mixed EX4200 and EX4500 Virtual Chassis Topology (Preprovisioned Configuration)	79
	Figure 12: Mixed EX4200 and EX4500 Virtual Chassis Topology (Nonprovisioned Configuration)	83
	Figure 13: Default Configuration of a Multimember EX4200 Virtual Chassis in a Single Wiring Closet	87
	Figure 14: EX4200 Virtual Chassis Interconnected Across Wiring Closets	94
	Figure 15: Mixed EX4200 and EX4500 Virtual Chassis Interconnected Across Wiring Closets	101
	Figure 16: Maximum Size EX4200 Virtual Chassis Interconnected Across Wiring Closets	110
	Figure 17: Topology for LAGs Connecting an EX4200 Virtual Chassis Access Switch to an EX4200 Virtual Chassis Distribution Switch	122
	Figure 18: Traffic Redirected by Fast Failover After VCP Link Failures in a Topology with Multiple Rings	132
	Figure 19: EX4200 Virtual Chassis Interconnected Across Wiring Closets to Form LAGs	138

List of Tables

	About the Documentation	xv
	Table 1: Notice Icons	xvii
	Table 2: Text and Syntax Conventions	xvii
Part 1	Overview	
Chapter 1	Virtual Chassis Overview	3
	Table 3: Minimum Junos OS Release by Virtual Chassis Connection Type	4
	Table 4: Maximum Member Switch Support for Virtual Chassis by Junos OS Release	10
	Table 5: Virtual Chassis Fabric Summary	17
	Table 6: Virtual Chassis Summary	18
	Table 7: Maximum Member Links in LAGs Over Optical Interface VCPs	27
	Table 8: Automatic Software Update Support	41
	Table 9: Effects of a Routing Engine Switchover	44
Part 2	Configuration	
Chapter 2	Configuration Examples	49
	Table 10: Components of the Basic Virtual Chassis Access Switch Topology	51
	Table 11: Components of the Basic Virtual Chassis Access Switch Topology	55
	Table 12: Components of the Basic Virtual Chassis Access Switch Topology	59
	Table 13: Components of the Basic Virtual Chassis Access Switch Topology	64
	Table 14: Components of the Expanded Virtual Chassis Access Switch	68
	Table 15: Components of the Expanded Virtual Chassis Access Switch	73
	Table 16: Components of the EX4200 Virtual Chassis Before the EX4500 Member Switches Are Added	78
	Table 17: Final Mixed EX4200 and EX4500 Virtual Chassis Components	78
	Table 18: Components of the EX4200 Virtual Chassis Before the EX4500 Member Switches Are Added	82
	Table 19: Final Mixed EX4200 and EX4500 Virtual Chassis Components	82
	Table 20: Components of a Virtual Chassis Interconnected Across Multiple Wiring Closets	93
	Table 21: Components of a Virtual Chassis Interconnected Across Wiring Closets	100
	Table 22: Components of a Preprovisioned EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets	108
	Table 23: Components of a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis	118
	Table 24: Components of the Topology for Connecting a Virtual Chassis Access Switch to a Virtual Chassis Distribution Switch	123

Chapter 3	Configuration Tasks	147
	Table 25: Virtual Chassis Configuration Fields	161
Part 3	Administration	
Chapter 5	Routine Monitoring	239
	Table 26: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration	240
Chapter 6	Operational Commands	253
	Table 27: request virtual chassis device reachability Output Fields	260
	Table 28: show snmp mib Output Fields	273
	Table 29: show system uptime Output Fields	277
	Table 30: show virtual-chassis active-topology Output Fields	280
	Table 31: show virtual-chassis device-topology Output Fields	285
	Table 32: show virtual-chassis protocol adjacency Output Fields	292
	Table 33: show virtual-chassis protocol database Output Fields	295
	Table 34: show virtual-chassis protocol interface Output Fields	300
	Table 35: show virtual-chassis protocol route Output Fields	302
	Table 36: show virtual-chassis protocol statistics Output Fields	305
	Table 37: show virtual-chassis fast-failover Output Fields	308
	Table 38: show virtual-chassis mode Output Fields	310
	Table 39: show virtual-chassis Output Fields	313
	Table 40: show virtual-chassis vc-path Output Fields	319
	Table 41: show virtual-chassis vc-port Output Fields	322
	Table 42: show virtual-chassis vc-port statistics Output Fields	327

About the Documentation

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

Documentation and Release Notes

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

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

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

Supported Platforms

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

- EX Series

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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


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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

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

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

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

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

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

Documentation Feedback

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

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

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

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

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

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

CHAPTER 1

Virtual Chassis Overview

- EX Series Virtual Chassis Overview on page 3
- Understanding EX2200 Virtual Chassis on page 8
- Understanding EX Series Virtual Chassis Components on page 9
- Understanding Mixed EX Series and QFX Series Virtual Chassis or Virtual Chassis Fabric on page 16
- Understanding How the Master in a Virtual Chassis Is Elected on page 20
- Understanding Software Upgrade in an EX Series Virtual Chassis on page 21
- Understanding Global Management of a Virtual Chassis on page 22
- Understanding Nonvolatile Storage in a Virtual Chassis on page 25
- Understanding the High-Speed Interconnection of the Dedicated Virtual Chassis Ports Connecting EX4200, EX4500, and EX4550 Member Switches on page 25
- Understanding EX Series Virtual Chassis Port Link Aggregation on page 26
- Understanding EX Series Virtual Chassis Configuration on page 28
- Understanding EX Series Virtual Chassis Switch Version Compatibility on page 30
- Understanding Fast Failover in an EX Series Virtual Chassis on page 30
- Understanding Split and Merge in a Virtual Chassis on page 37
- Understanding Automatic Software Update on Virtual Chassis Member Switches on page 40
- Understanding Virtual Chassis Device Reachability Testing on page 42
- Understanding MAC Address Assignment on a Virtual Chassis on page 43
- Understanding High Availability on an EX Series Virtual Chassis on page 44

EX Series Virtual Chassis Overview



NOTE: This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis. See *EX8200 Virtual Chassis Overview* for information about EX8200 Virtual Chassis.

Many Juniper Networks EX Series switches support the *Virtual Chassis* flexible, scaling switch solution. You can connect individual switches together to form one unit and manage the unit as a single chassis.

The following Virtual Chassis are supported on EX Series switches:

- An EX2200 Virtual Chassis, composed of up to four EX2200 switches
- An EX3300 Virtual Chassis, composed of up to ten EX3300 switches
- An EX4200 Virtual Chassis, composed of up to ten EX4200 switches
- An EX4300 Virtual Chassis, composed of up to ten EX4300 switches
- An EX4500 Virtual Chassis, composed of up to ten EX4500 switches
- An EX4550 Virtual Chassis, composed of up to ten EX4550 switches
- An EX4600 Virtual Chassis, composed of up to ten EX4600 switches.
- A mixed EX4200 and EX4500 Virtual Chassis, a Virtual Chassis composed of up to ten total EX4200 and EX4500 switches
- A mixed EX4200 and EX4550 Virtual Chassis, a Virtual Chassis composed of up to ten total EX4200 and EX4550 switches
- A mixed EX4200, EX4500, and EX4550 Virtual Chassis, a Virtual Chassis composed of up to ten total EX4200, EX4500, and EX4550 switches
- A mixed EX4300 and EX4600 Virtual Chassis, a Virtual Chassis composed of up to ten total EX4300 and EX4600 switches.
- A mixed EX4500 and EX4550 Virtual Chassis, a Virtual Chassis composed of up to ten total EX4500 and EX4550 switches
- An EX8200 Virtual Chassis, composed of up to four EX8200 switches



NOTE: This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis. See *EX8200 Virtual Chassis Overview*.

Table 3 on page 4 lists the initial Junos OS release that supports each Virtual Chassis combination.

Table 3: Minimum Junos OS Release by Virtual Chassis Connection Type

Switch	EX2200 Switch	EX3300 Switch	EX4200 Switch	EX4300 Switch	EX4500 Switch	EX4550 Switch	EX4600 Switch
EX2200 Switch	12.2R1	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
EX3300 Switch	Not supported	11.3R1	Not supported	Not supported	Not supported	Not supported	Not supported
EX4200 Switch	Not supported	Not supported	9.0R1	Not supported	11.1R1	12.2R1	Not supported

Table 3: Minimum Junos OS Release by Virtual Chassis Connection Type (continued)

Switch	EX2200 Switch	EX3300 Switch	EX4200 Switch	EX4300 Switch	EX4500 Switch	EX4550 Switch	EX4600 Switch
EX4300 Switch	Not supported	Not supported	Not supported	13.2X51-D25	Not supported	Not supported	13.2X51-D25
EX4500 Switch	Not supported	Not supported	11.1R1	Not supported	11.1R1	12.2R1	Not supported
EX4550 Switch	Not supported	Not supported	12.2R1	Not supported	12.2R1	12.2R1	Not supported
EX4600 switch	Not supported	Not supported	Not supported	13.2X51-D25	Not supported	Not supported	13.2X51-D25

This topic describes:

- [Basic Configuration of a Virtual Chassis on page 5](#)
- [Expanding Configurations—Within a Single Wiring Closet and Across Wiring Closets on page 6](#)
- [Global Management of Member Switches in a Virtual Chassis on page 7](#)
- [High Availability Through Redundant Routing Engines on page 7](#)
- [Adaptability as an Access Switch or Distribution Switch on page 7](#)

Basic Configuration of a Virtual Chassis

You need to interconnect at least two switches to form a Virtual Chassis.

You interconnect EX2200 switches into an EX2200 Virtual Chassis by configuring uplink ports that connect two or more switches as Virtual Chassis Ports (VCPs). You can configure all EX2200 uplink ports which support 1-Gbps speeds into VCPs.

You interconnect EX3300 switches into a Virtual Chassis by interconnecting uplink port connections configured as VCPs between two or more switches. By default, uplink ports 2 and 3 on EX3300 switches are configured as VCPs. EX3300 switches do not have dedicated VCPs. See [“Configuring an EX3300 Virtual Chassis \(CLI Procedure\)” on page 152](#).

You interconnect EX4200, EX4500, and EX4550 switches in a Virtual Chassis by either interconnecting the switches through the dedicated VCPs or by configuring the optical port connections as VCPs. All EX4200 switches are shipped with two built-in dedicated VCPs. The dedicated VCPs are on the Virtual Chassis module in an EX4500 or EX4550 switch. All supported SFP, SFP+, and XFP uplink connections between EX4200, EX4500, and EX4550 switches can be configured as VCPs.

You interconnect EX4300 switches and EX4600 switches into a Virtual Chassis using the uplink ports. All uplink ports on EX4300 and EX4600 switches are not configured into VCPs by default, with the exception that all 40-Gbps QSFP+ ports on an EX4300 switch are configured as VCPs by default.

See [“Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\)”](#) on page 156 or [“Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\)”](#) on page 165.

Expanding Configurations—Within a Single Wiring Closet and Across Wiring Closets

Within a single wiring closet, you can simply add an EX4200, EX4500, or EX4550 member switch to a Virtual Chassis by cabling the member switch into the Virtual Chassis using the dedicated VCPs.

You can easily expand a Virtual Chassis configuration beyond a single wiring closet or over a longer distance by connecting member switches together using SFP, SFP+, or XFP uplink connections and configuring those connections as VCPs. All supported SFP, SFP+, and XFP uplink connections between EX4200, EX4500, and EX4550 switches can be configured as VCPs. All supported SFP+ and QSFP+ uplink connections between EX4300 and EX4600 switches can also be configured into VCPs.

EX2200 and EX3300 switches do not have dedicated VCPs. You must always use the uplink ports on an EX2200 or an EX3300 switch to connect the Virtual Chassis both within a wiring closet and across wiring closets. Uplink ports 2 and 3 are automatically configured as VCPs on EX3300 switches. No uplink ports on EX2200 switches are automatically configured as VCPs.

When you are creating a Virtual Chassis, you might want to deterministically control the role and member ID assigned to each member switch. You can do this by creating a preprovisioned configuration. You can add switches to a preprovisioned configuration by using the autoprovisioning feature to automatically configure the uplink ports as VCPs on the switches being added.

To add a new switch to a wiring closet, see:

- [Adding a New Switch to an Existing EX3300 Virtual Chassis \(CLI Procedure\)](#) on page 169
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\)](#) on page 171
- [Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\)](#) on page 176
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\)](#) on page 178
- [Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\)](#) on page 180

To manually configure a VCP, see:

- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\)](#) on page 188
- [Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\)](#) on page 193

To use preprovisioning to configure a Virtual Chassis or autoprovisioning to add a switch to a Virtual Chassis, see:

- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\) on page 171](#)

Global Management of Member Switches in a Virtual Chassis

The interconnected member switches in a Virtual Chassis operate as a single network entity. You run EZSetup only once to specify the identification parameters for the master, and these parameters implicitly apply to all members of the Virtual Chassis. You can view the Virtual Chassis as a single device in the J-Web user interface (on platforms and software that support J-Web) and apply various device management functions to all members of the Virtual Chassis.

The serial console port and dedicated out-of-band management port that are on the rear panel of the individual switches have global virtual counterparts when the switches are interconnected in a Virtual Chassis configuration. A PC or laptop allows you to connect to the master 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. See [“Understanding Global Management of a Virtual Chassis” on page 22](#).

High Availability Through Redundant Routing Engines

You increase your network's high availability when you interconnect an EX Series 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.

A Virtual Chassis has dual Routing Engines—the switch in the master role and the switch in the backup role—and therefore supports many high availability features not supported on standalone EX Series switches. See [“Understanding High Availability on an EX Series Virtual Chassis” on page 44](#).

You can further improve the high availability of your network by configuring the high availability 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.

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 models 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 the 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 configure one more Virtual Chassis configuration with partial PoE models to support the company's external servers. Alternatively, you can use the Virtual Chassis as a distribution switch.

**Related
Documentation**

- [EX8200 Virtual Chassis Overview](#)
- [Understanding EX Series Virtual Chassis Components on page 9](#)
- [Understanding How the Master in a Virtual Chassis Is Elected on page 20](#)
- [Understanding EX Series Virtual Chassis Switch Version Compatibility on page 30](#)
- [Understanding EX Series Virtual Chassis Port Link Aggregation on page 26](#)
- [Understanding EX Series Virtual Chassis Configuration on page 28](#)

Understanding EX2200 Virtual Chassis

EX2200 Virtual Chassis brings the Virtual Chassis flexible, scaling switch solution to the Juniper Networks EX2200 Ethernet Switch. You can connect up to four EX2200 switches together to form one EX2200 Virtual Chassis and manage the unit as a single chassis. The advantages of connecting multiple switches into a Virtual Chassis include better-managed bandwidth at a network layer, simplified configuration and maintenance because multiple devices can be managed as a single device, increased fault tolerance and high availability (HA) because a Virtual Chassis can remain active and network traffic can be redirected to other member switches when a single member switch fails, and a simplified Layer 2 network topology that minimizes or eliminates the need for loop prevention protocols such as Spanning Tree Protocol (STP).

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. 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. EX2200 and EX2200-C switches can be interconnected into the same EX2200 Virtual Chassis. EX2200 switches cannot be interconnected into a Virtual Chassis with any other Juniper Networks EX Series Ethernet switches.

An EX2200 Virtual Chassis is configured, monitored, and maintained like other EX Series Virtual Chassis. See [“Understanding EX Series Virtual Chassis Components” on page 9](#).

An EX2200 Virtual Chassis has the following limitations that are not applicable to other EX Series Virtual Chassis:

- Support for only four member switches.
- Limited VCP bandwidth to transport network traffic across the Virtual Chassis. EX2200 switches support only 1 Gbps interfaces, and configuring these 1 Gbps interfaces into VCPs is the only way to interconnect EX2200 switches into an EX2200 Virtual Chassis. You can configure up to 8 1 Gbps interfaces configured as VCPs into a single Link

Aggregation Group (LAG) bundle, so a VCP LAG supports a maximum bandwidth of 8 Gbps.

On other EX Series Virtual Chassis, you can configure optical ports that support 10 Gbps as VCPs. These 10 Gbps ports form VCP LAGs that can have significantly higher overall bandwidths. See [“Understanding EX Series Virtual Chassis Port Link Aggregation” on page 26](#).

The lowest bandwidth for a dedicated VCP used to interconnect an EX Series Virtual Chassis is 64 Gbps.

- Limited support for high availability features. Most high availability features, including Graceful Routing Engine switchover (GRES), Nonstop bridging (NSB), Nonstop active routing (NSR), and Nonstop software upgrade (NSSU), are not supported on an EX2200 Virtual Chassis.
- No support for fast failover.

**Related
Documentation**

- [EX Series Virtual Chassis Overview on page 3](#)
- [Configuring an EX2200 Virtual Chassis \(CLI Procedure\) on page 148](#)

Understanding EX Series Virtual Chassis Components



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

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

This topic covers:

- [Maximum Number of Switches per Virtual Chassis on page 10](#)
- [Virtual Chassis Ports \(VCPs\) on page 11](#)
- [Master Role on page 12](#)
- [Backup Role on page 13](#)
- [Linecard Role on page 13](#)
- [Member Switch and Member ID on page 14](#)
- [Mastership Priority on page 14](#)
- [Mixed Virtual Chassis on page 15](#)
- [Virtual Chassis Identifier \(VCID\) on page 15](#)

Maximum Number of Switches per Virtual Chassis

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

[Table 4 on page 10](#) lists the maximum member switch support by Virtual Chassis and Junos OS release.

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

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

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

Maximum Member Switch Support	Initial Junos OS Release
Mixed EX4200, EX4500, and EX4550 Virtual Chassis	12.2R1—Initial release. Support for up to ten total EX4200, EX4500, and EX4550 switches
Mixed EX4300 and EX4600 Virtual Chassis	13.2X51-D25—Initial release. Support for up to ten total EX4300 and EX4600 switches. EX4600 switches must assume routing engine role.
Mixed EX4500 and EX4550 Virtual Chassis	12.2R1—Initial release. Support for up to ten total EX4500 and EX4550 switches
EX9200 Virtual Chassis	13.2R2—Initial release. Support for up to two EX9200 switches.

Virtual Chassis Ports (VCPs)

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

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

These switches have dedicated VCPs:

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

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

- EX2200 switches, through an uplink port



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

- EX3300 switches, through an uplink port



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

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



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

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

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

You can increase the Virtual Chassis bandwidth between member switches by configuring multiple optical ports connecting the switches as VCPs. The optical ports configured as VCPs automatically form a Link Aggregation Group (LAG) bundle. See [“Understanding EX Series Virtual Chassis Port Link Aggregation”](#) on page 26.

Master Role

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

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

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

- In a preprovisioned configuration, one of the two members assigned as **routing-engine** functions as the master member. The selection of which member assigned as **routing-engine** functions as master and which as backup is determined by the software based on the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected”](#) on page 20.
- In a configuration that is not preprovisioned, the selection of the master and backup is determined by the mastership priority value and secondary factors in the master election algorithm.

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

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

Backup Role

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

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

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

- In a preprovisioned configuration, one of the two members assigned as **routing-engine** functions in the backup role. The selection of which member assigned as **routing-engine** functions as master and which as backup is determined by the software based on the master election algorithm. See ["Understanding How the Master in a Virtual Chassis Is Elected" on page 20](#).
- In a configuration that is not preprovisioned, the selection of the master and backup is determined by the mastership priority value and secondary factors in the master election algorithm.

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

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

Linecard Role

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

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

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

- In a preprovisioned configuration, you can explicitly configure a member with the linecard role, which makes it ineligible for functioning as a master or backup.

- In a configuration that is not preprovisioned, the members that are not selected as master or backup function as linecard members of the Virtual Chassis configuration. The selection of the master and backup is determined by the mastership priority value and secondary factors in the master election algorithm. A switch with a mastership priority of 0 is always in the linecard role.

Member Switch and Member ID

Each standalone EX Series switch that supports Virtual Chassis is a potential member of a Virtual Chassis configuration. When one of those switches is powered on, it receives a member ID that can be seen by viewing the front-panel LCD or by entering the **show virtual-chassis** command. If the switch is powered on as a standalone switch, that member's member ID is always 0. When the switch is interconnected with other switches in a Virtual Chassis configuration, its member ID is assigned by the master based on various factors, such as the order in which the switch was added to the Virtual Chassis configuration or the member ID assigned by a preprovisioned configuration. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#).

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

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

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

Mastership Priority

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

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



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

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

Mixed Virtual Chassis

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

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

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

Virtual Chassis Identifier (VCID)

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

Related Documentation

- [Understanding EX8200 Virtual Chassis Components](#)
- [EX Series Virtual Chassis Overview on page 3](#)
- [Understanding EX4300 Virtual Chassis](#)
- [Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58](#)
- [Example: Configuring an EX4500 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 63](#)

- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 188](#)
- [Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) on page 193](#)

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

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

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

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



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

This topic covers:

- [Virtual Chassis Fabric Summary on page 16](#)
- [Understanding Mixed Virtual Chassis Fabric on page 17](#)
- [Virtual Chassis Summary for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 Switches on page 17](#)
- [Understanding the Routing Engine Role in a Mixed Virtual Chassis Using EX4300, EX4600, QFX3500, QFX3600, or QFX5100 Member Switches on page 18](#)
- [Understanding EX4300, QFX3500, QFX3600, and QFX5100 Switches in a Virtual Chassis on page 19](#)
- [Understanding Mixed EX4300 and EX4600 Virtual Chassis on page 19](#)
- [Understanding EX4200, EX4500, and EX4550 Switches in a Mixed Virtual Chassis on page 19](#)

[Virtual Chassis Fabric Summary](#)

[Table 5 on page 17](#) provides a high-level overview of the permitted hardware allowed in the routing engine and line card roles of a mixed and a non-mixed VCF. The table also includes license requirements and supported configuration methods.

Table 5: Virtual Chassis Fabric Summary

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

Understanding Mixed Virtual Chassis Fabric

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

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

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

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

Table 6 on page 18 provides a high-level overview of the permitted hardware allowed in the routing engine and line card roles of a mixed and a non-mixed Virtual Chassis for QFX5100, QFX3600, QFX3500, EX4600, and EX4300 switches. The table also includes license requirements and supported configuration methods.

Table 6: Virtual Chassis Summary

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

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

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

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

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

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

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

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

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

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

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

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

Understanding Mixed EX4300 and EX4600 Virtual Chassis

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

The mixed Virtual Chassis supports up to ten member switches.

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

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

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

Related Documentation

- *Virtual Chassis Fabric Overview*
- *Understanding QFX Series Virtual Chassis*
- [EX Series Virtual Chassis Overview on page 3](#)
- *Understanding Virtual Chassis Fabric Components*

- [Understanding QFX Series Virtual Chassis Components](#)
- [Understanding EX Series Virtual Chassis Components on page 9](#)

Understanding How the Master in a Virtual Chassis Is Elected



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

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

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

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

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

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

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

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

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

Procedure)” on page 152, “Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure)” on page 156, or *Configuring a QFX Series Virtual Chassis (CLI Procedure)*.

Related Documentation

- *EX8200 Virtual Chassis Overview*
- [EX Series Virtual Chassis Overview on page 3](#)
- *Understanding QFX Series Virtual Chassis*
- [Understanding EX Series Virtual Chassis Components on page 9](#)
- *Understanding QFX Series Virtual Chassis Components*
- [Understanding EX Series Virtual Chassis Configuration on page 28](#)

Understanding Software Upgrade in an EX Series Virtual Chassis



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

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

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

You can use the automatic software update feature to automatically update the Junos OS version on member switches as you add them to a Virtual Chassis. See “[Understanding Automatic Software Update on Virtual Chassis Member Switches](#)” on page 40. If you are not configuring the automatic software update feature, we recommend that you update the new member switch to the version of Junos OS running on the Virtual Chassis before adding the member switch to the Virtual Chassis.

In any mixed Virtual Chassis, the member switches must be running the same version of Junos OS. You can upgrade all member switches simultaneously by specifying a path to multiple Junos OS images in the same **request system software add** command. Multiple Junos OS images are needed to upgrade a mixed Virtual Chassis because an EX4200 switch runs a different version of Junos OS than an EX4500 or EX4550 switch. See “[Installing Software on a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Switches \(CLI Procedure\)](#)” on page 168.



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

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

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

**Related
Documentation**

- [Understanding Software Upgrades in an EX8200 Virtual Chassis](#)
- [Understanding EX Series Virtual Chassis Components on page 9](#)
- [Understanding Automatic Software Update on Virtual Chassis Member Switches on page 40](#)
- [Installing Software on an EX Series Switch with a Single Routing Engine \(CLI Procedure\)](#)
- [Upgrading Software on an EX3300, EX4200, EX4300, EX4500 and EX4550 Virtual Chassis, and Mixed Virtual Chassis Using Nonstop Software Upgrade \(CLI Procedure\)](#)

Understanding Global Management of a Virtual Chassis

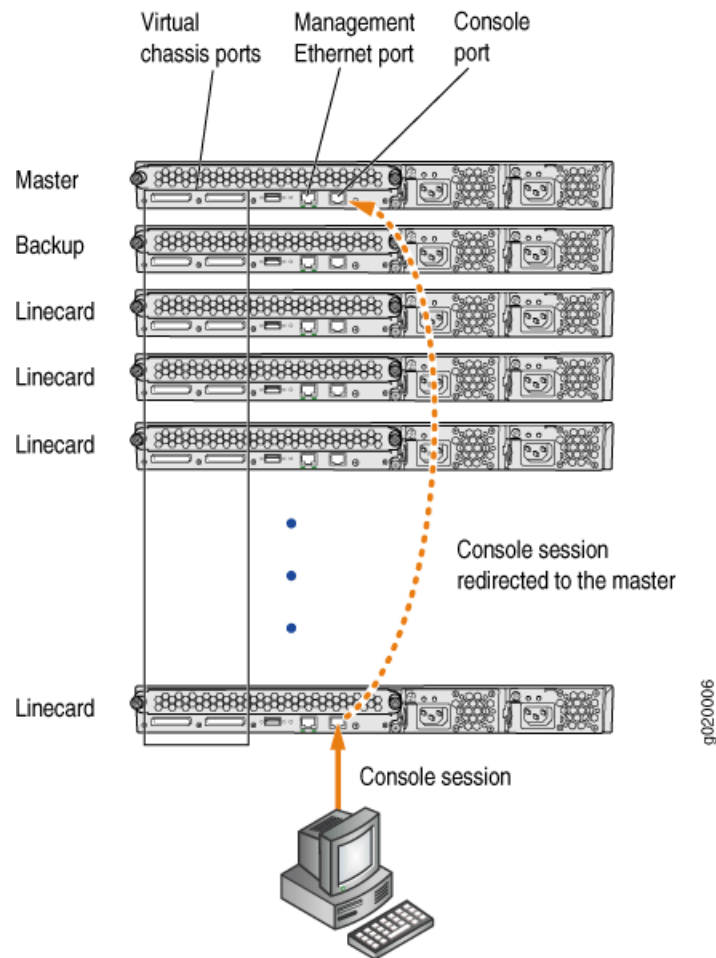


NOTE: This topic does not apply to EX8200 Virtual Chassis. See *Understanding Global Management of an EX8200 Virtual Chassis*.

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

You can connect a PC or laptop directly to a console port of any member switch to set up and configure the Virtual Chassis. When you connect to the console port of any member switch, the console session is redirected to the master switch, as shown in [Figure 1 on page 23](#).

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

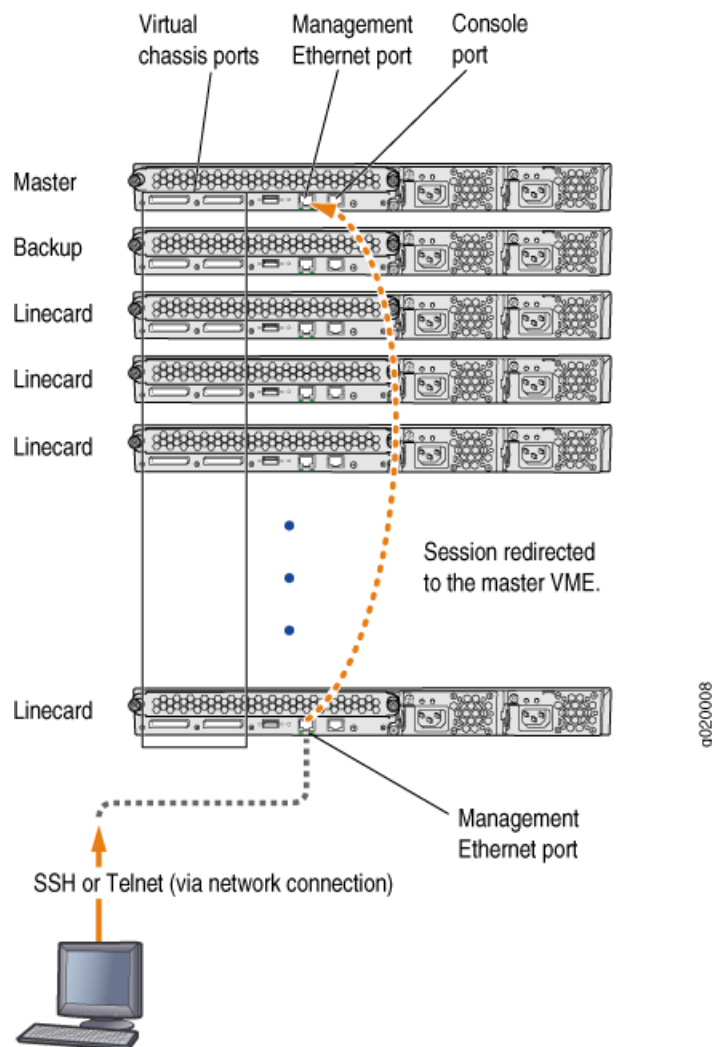


If the master becomes unavailable, the console session is disconnected from the old master and a new session is established with the newly elected master.

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

The Virtual Chassis configuration can be managed remotely through SSH or Telnet using a global management interface called the virtual management Ethernet (VME) interface. The VME interface is a logical interface representing all of the out-of-band management ports on the member switches. When you connect to the Virtual Chassis configuration using the VME interface's IP address, the connection is redirected to the master member as shown in [Figure 2 on page 24](#).

Figure 2: Management Ethernet Port Redirection to the VME Interface



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

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

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

Related Documentation

- [Understanding Global Management of an EX8200 Virtual Chassis](#)
- [Understanding EX Series Virtual Chassis Components on page 9](#)
- [Understanding QFX Series Virtual Chassis Components](#)

- [Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58](#)
- [Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\) on page 195](#)

Understanding Nonvolatile Storage in a Virtual Chassis



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

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

- [Nonvolatile Memory Features on page 25](#)

Nonvolatile Memory Features

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

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

Related Documentation

- [Understanding File Storage in an EX8200 Virtual Chassis](#)
- [Understanding QFX Series Virtual Chassis](#)
- [Command Forwarding Usage with an EX Series Virtual Chassis on page 239](#)

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 EX Series Virtual Chassis Components on page 9](#)
- [Virtual Chassis Cabling Configuration Examples for EX4200 Switches](#)

Understanding EX Series Virtual Chassis Port Link Aggregation



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

You can combine physical Ethernet ports belonging to different member switches of a Virtual Chassis configuration to form a logical point-to-point link, known as a *link aggregation group (LAG) or bundle*. A LAG provides more bandwidth than a single Ethernet link can provide. Additionally, link aggregation provides network redundancy by load-balancing traffic across all available links. If one of the links fails, the system automatically load-balances traffic across all remaining links.

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

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

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

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

On an EX2200 switch only, you can also configure all RJ-45 interfaces, including built-in network ports with 10/100/1000BASE-T Gigabit Ethernet connectors and 1000BASE-T

RJ-45 transceivers, on EX2200 and EX2200-C switches, into VCPs. The RJ-45 interfaces also automatically form a LAG when configured into VCPs.

[Table 7 on page 27](#) provides the maximum member link limit for each optical VCP LAG.

Table 7: Maximum Member Links in LAGs Over Optical Interface VCPs

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

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

See [“Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\)” on page 188](#) for information about configuring uplink ports as VCPs.



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

Related Documentation

- [Understanding Virtual Chassis Port Link Aggregation in an EX8200 Virtual Chassis](#)
- [EX Series Virtual Chassis Overview on page 3](#)
- [Understanding Aggregated Ethernet Interfaces and LACP](#)
- [Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch on page 120](#)

- [Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch on page 126](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets on page 91](#)
- [Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets on page 98](#)
- [Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports on page 136](#)

Understanding EX Series Virtual Chassis Configuration



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

You configure and manage almost all aspects of an EX Series Virtual Chassis configuration through the master 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.

EX Series switches that support Virtual Chassis have some innate characteristics of a Virtual Chassis by default. A standalone switch is assigned member ID 0 and is the master of itself. Therefore, you can edit its Virtual Chassis configuration. When the standalone switch is interconnected with an existing Virtual Chassis configuration, the Virtual Chassis configuration statements and any uplink Virtual Chassis port (VCP) settings that you previously specified on the standalone switch remain part of its configuration.

A switch is not recognized by the Virtual Chassis as a member switch until it is interconnected with the master or interconnected with an existing member of the Virtual Chassis.

EX2200, EX3300, and EX4300 switches do not have dedicated VCPs. These switches form a Virtual Chassis by configuring optical ports connecting the member switches into VCPs. You can expand the uplink connections on these switches over long distances to connect switches at different locations into the same Virtual Chassis.



NOTE: Uplink ports 2 and 3 on any EX3300 switch are configured, by default, as VCPs. You can change this default configuration or configure another uplink port as a VCP on an EX3300 switch.

QSFP+ ports on an EX4300 switch are also configured, by default, as VCPs.

On all other EX Series switches, the optical ports are not configured as VCPs by default.

When an EX4200, EX4500, or EX4550 switch is located too far away to be interconnected through dedicated VCPs, you can configure an optical port as a VCP.

A link aggregation group (LAG) will be formed automatically when the new switch is added to the configuration if more than one such link with the same speed is detected between optical VCPs on the new member and an existing member. See [“Understanding EX Series Virtual Chassis Port Link Aggregation” on page 26](#).

When an optical port is set as a VCP, it cannot be used for any other purpose. If you want to use the optical port for another purpose, you must delete the VCP setting. You can execute this command directly on the member whose uplink VCP setting you want to delete or through the master 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” on page 37](#).

You can create a preprovisioned configuration. This type of configuration allows you to deterministically control the member ID and role assigned to a member switch by associating the switch with its serial number. For an example of a preprovisioned configuration, see [“Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File” on page 104](#).



NOTE: If a switch is interconnected with other switches in a Virtual Chassis configuration, each individual switch that is included as a member of the configuration is identified with a member ID. The member ID 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.

Related Documentation

- [Understanding EX8200 Virtual Chassis Components](#)
- [Understanding EX Series Virtual Chassis Components on page 9](#)
- [Understanding How the Master in a Virtual Chassis Is Elected on page 20](#)
- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) on page 152](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)

Understanding EX Series Virtual Chassis Switch Version Compatibility



NOTE: This topic applies to all EX Series Virtual Chassis except EX8200 Virtual Chassis. See *Understanding EX8200 Virtual Chassis Compatibility Requirements* for information about EX8200 Virtual Chassis.

EX2200, EX3300, EX4200, EX4300, EX4500, or EX4550 switches must be running the same software versions to be interconnected into a Virtual Chassis. The master checks the hardware version, the Juniper Networks Junos operating system (Junos OS) version, and other component versions running in a switch that is physically interconnected to its Virtual Chassis port (VCP). Different hardware models can be members of the same Virtual Chassis configuration, and any combination of EX4200, EX4500, or EX4550 switches can be connected together to form a mixed Virtual Chassis. However, the master will not assign a member ID to a switch that is running a different software version. A switch that is running a different version of software cannot join the Virtual Chassis configuration.

Related Documentation

- [Understanding EX8200 Virtual Chassis Compatibility Requirements](#)
- [Understanding EX Series Virtual Chassis Components on page 9](#)
- [Understanding Software Upgrade in an EX Series Virtual Chassis on page 21](#)
- [Understanding Software Installation on EX Series Switches](#)
- [Installing Software on an EX Series Switch with a Single Routing Engine \(CLI Procedure\)](#)

Understanding Fast Failover in an EX Series 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 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 on page 31](#)
- [How Fast Failover Works on page 31](#)
- [Effects of Topology Changes on a Fast Failover Configuration on page 36](#)

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 32](#) shows normal traffic flow in a ring topology using dedicated VCPs.

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

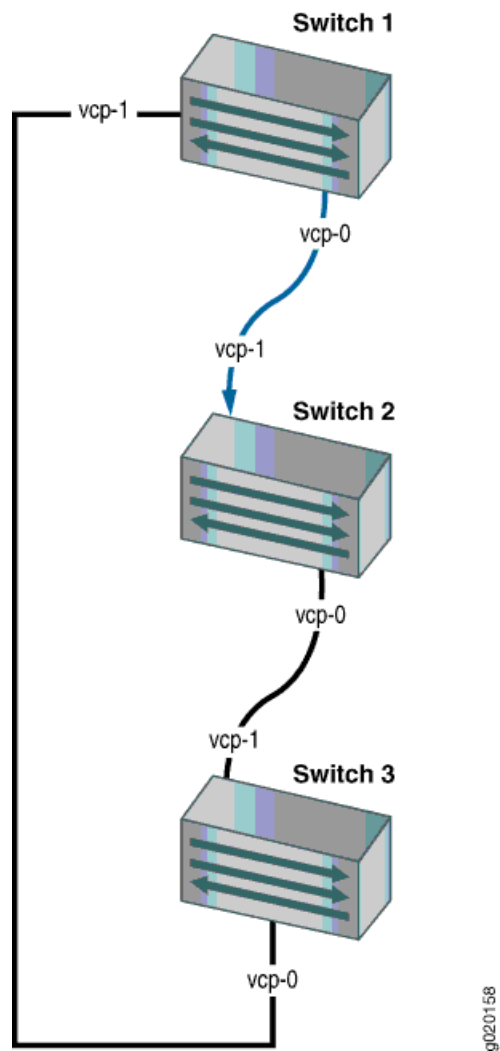
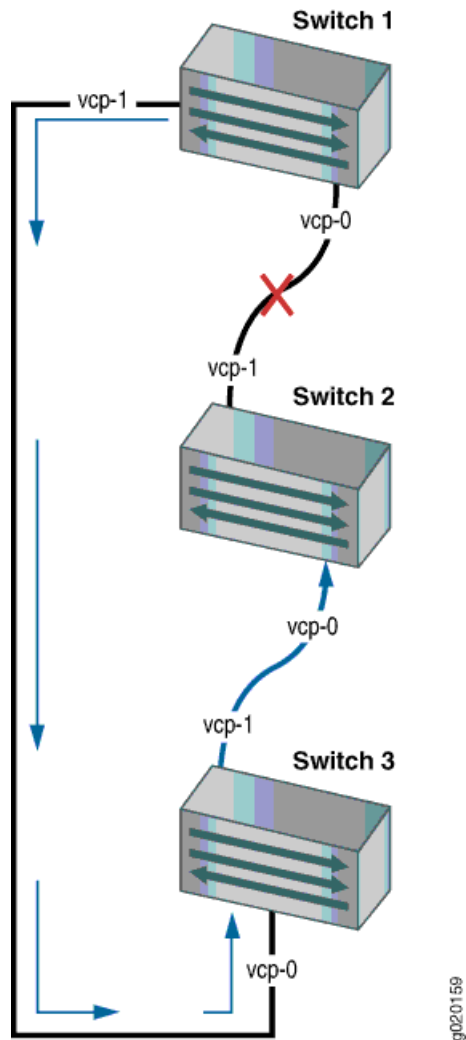


Figure 4 on page 33 shows traffic redirected by fast failover.

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 34](#) 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

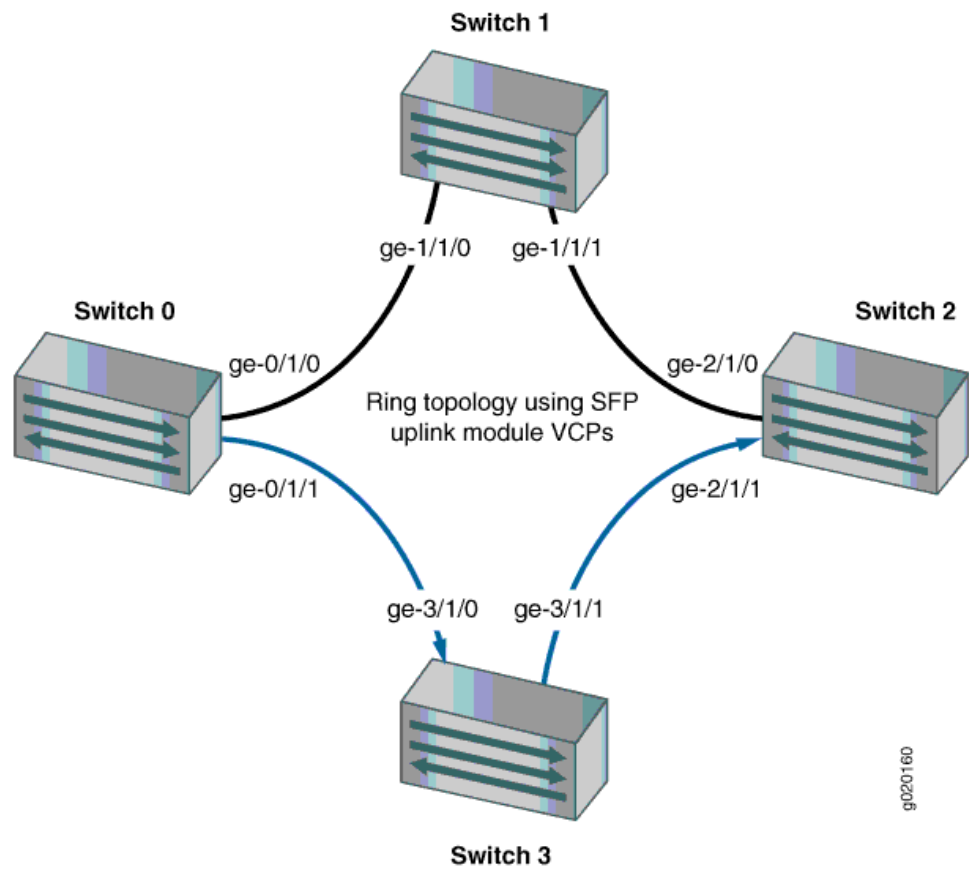
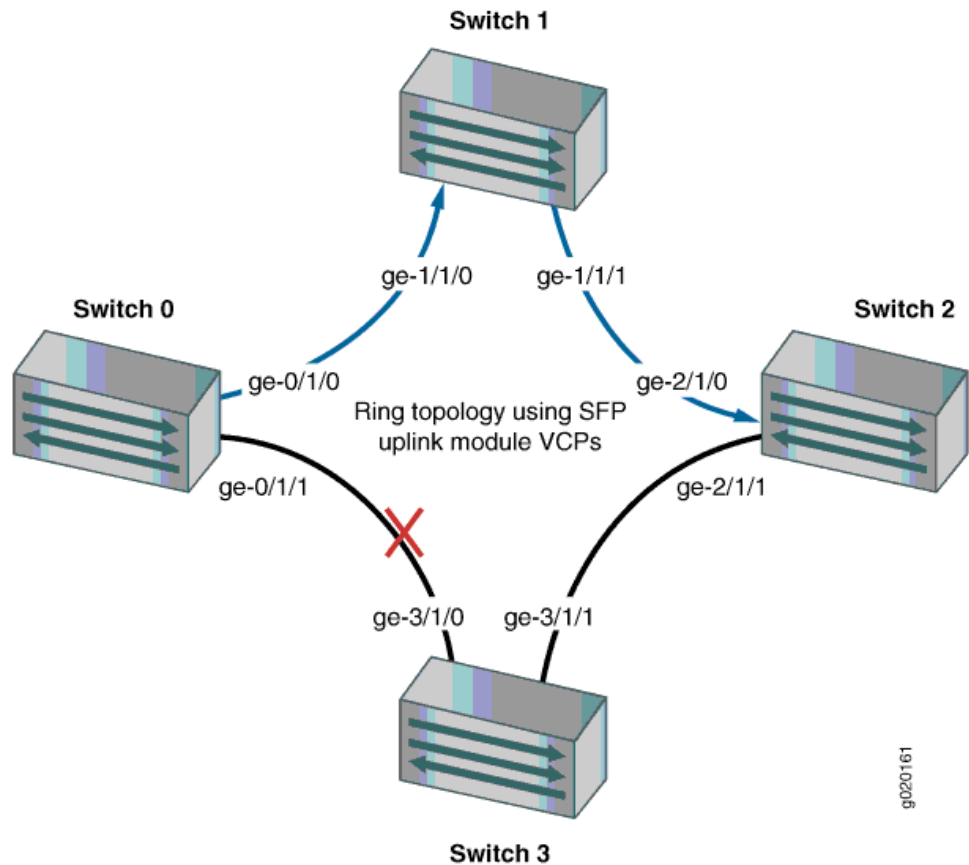


Figure 6 on page 35 shows traffic redirected by fast failover.

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



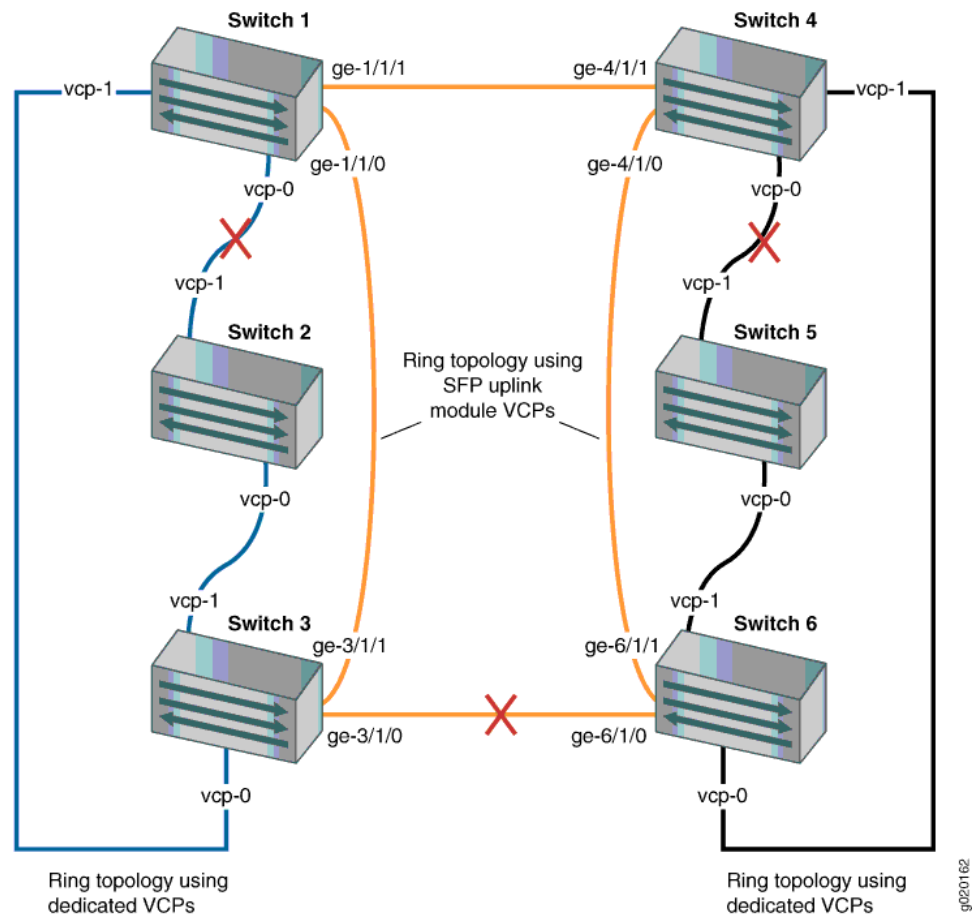
In a ring topology that uses SFP uplink VCPs, there are four ports per module. Consecutive pairs of ports are automatically configured as backup ports for each other. For example, if a Virtual Chassis member has an SFP uplink module installed, uplink module VCPs `ge-0/1/0` and `ge-0/1/1` are automatically 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 36](#).

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 Configuration on page 28](#)
- [Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails on page 131](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 188](#)

Understanding Split and Merge in a Virtual Chassis

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



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

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



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

This topic describes:

- [What Happens When a Virtual Chassis Configuration Splits on page 37](#)
- [Merging Virtual Chassis Configurations on page 38](#)

What Happens When a Virtual Chassis Configuration Splits

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

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

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



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

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

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



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

Merging Virtual Chassis Configurations

There are two scenarios in which separate Virtual Chassis merge:

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

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

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

- If the Virtual Chassis configurations have the same VCID, then the configurations can merge. If the two Virtual Chassis were formed as the result of a split, they have the same VCID.
- If the VCIDs are different, then the two configurations can merge only if both are active (inactive configurations cannot merge, ensuring that members removed from one Virtual Chassis configuration do not become members of another Virtual Chassis

configuration). If the configurations to merge are both active and one of them has a user-configured VCID, this ID becomes the ID of the merged Virtual Chassis. If neither Virtual Chassis has a user-configured VCID, then the VCID of the configuration with the highest mastership priority becomes the ID of the merged Virtual Chassis. The resulting merged Virtual Chassis configuration is active.

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

1. Connecting the two split Virtual Chassis configurations triggers the shortest-path-first (SPF) algorithm. The SPF algorithm computes the network topology and then triggers the master election algorithm. The master election algorithm waits for the members to synchronize the topology information before running.
2. The master election algorithm merges the VCIDs of all the members.
3. Each member runs the master election algorithm to select a master and a backup from among all members with the same VCIDs. For more information, see [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#).
4. The master determines whether the Virtual Chassis configuration is active or inactive. (See [“What Happens When a Virtual Chassis Configuration Splits” on page 37](#).)
5. If the Virtual Chassis configuration is active, the master assigns roles to all members. If the Virtual Chassis configuration is inactive, the master assigns all members the role of linecard.
6. When the other members receive their role from the master, they change their role to backup or linecard. They also use the active or inactive state information sent by the master to set their own state to active or inactive and to construct the Virtual Chassis member list from the information sent by the master.
7. If the Virtual Chassis state is active, the master waits for messages from the members indicating that they have changed their roles to the assigned roles, and then the master changes its own role to master.



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

Related Documentation

- [Disabling Split and Merge in a Virtual Chassis \(CLI Procedure\) on page 199](#)
- [Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge \(CLI Procedure\) on page 201](#)
- [Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge on page 134](#)
- [Understanding EX Series Virtual Chassis Configuration on page 28](#)
- [Understanding QFX Series Virtual Chassis](#)

Understanding Automatic Software Update on Virtual Chassis Member Switches

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

This topic includes:

- [Automatic Software Update Basics on page 40](#)
- [Automatic Software Update Restrictions on page 40](#)

Automatic Software Update Basics

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

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

Automatic Software Update Restrictions

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

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

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

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

[Table 8 on page 41](#) summarizes automatic software update support for each Junos OS release combination.

Table 8: Automatic Software Update Support

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

Table 8: Automatic Software Update Support (*continued*)

Virtual Chassis Junos OS Release	Supported Junos OS Releases for Prospective Member Switches
Junos OS Releases 10.4R1 through 10.4R3	All versions of Junos OS 9.0 through 9.6 Junos OS Releases 10.0R1 through 10.0R4 All versions of Junos OS Release 10.1 Junos OS Releases 10.2R1 through 10.2R3 Junos OS Releases 10.3R1 through 10.3R3
Junos OS Release 10.4R4 and later 10.4 releases	Junos OS Release 10.0R5 Junos OS Release 10.2R4 and later 10.2 releases Junos OS Release 10.3R4 and later 10.3 releases All versions of Junos OS Release 11.1
Junos OS Release 11.1R1	All versions of Junos OS Release 10.4 Junos OS Release 11.2 and later Junos OS releases
Junos OS Release 11.1R2 and later Junos OS releases	Junos OS Release 10.0R5 Junos OS Release 10.2R4 and later 10.2 releases Junos OS Release 10.3R4 and later 10.3 releases Junos OS Release 11.2 and later Junos OS releases

Related Documentation

- [Understanding Software Upgrade in an EX Series Virtual Chassis on page 21](#)
- [Understanding Software Upgrades in a QFX Series Virtual Chassis](#)
- [Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches on page 144](#)
- [Configuring Automatic Software Update on Virtual Chassis Member Switches \(CLI Procedure\) on page 199](#)

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

- [Verifying Connectivity Between Virtual Chassis Member Devices on page 251](#)

Understanding MAC Address Assignment on a Virtual Chassis

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

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

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

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



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

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

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

Related Documentation

- [Understanding MAC Address Assignment in an EX Series Switch](#)
- [Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis \(CLI Procedure\) on page 196](#)
- [EX Series Virtual Chassis Overview on page 3](#)

- *EX8200 Virtual Chassis Overview*
- *Understanding QFX Series Virtual Chassis*

Understanding High Availability on an EX Series Virtual Chassis

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

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

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

Many HA features for the EX Series Virtual Chassis are designed to improve network resiliency after a Routing Engine switchover. [Table 9 on page 44](#) describes the effects of a Routing Engine switchover when no high availability features are enabled and when some High Availability features are enabled.

Table 9: Effects of a Routing Engine Switchover

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

Table 9: Effects of a Routing Engine Switchover (*continued*)

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

- Related Documentation**
- [EX Series Virtual Chassis Overview on page 3](#)
 - [High Availability Features for EX Series Switches Overview](#)

PART 2

Configuration

- [Configuration Examples on page 49](#)
- [Configuration Tasks on page 147](#)
- [Configuration Statements on page 205](#)

CHAPTER 2

Configuration Examples

- [Example: Configuring an EX2200 Virtual Chassis with Four Member Switches on page 50](#)
- [Example: Configuring an EX3300 Virtual Chassis with a Master and Backup on page 54](#)
- [Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58](#)
- [Example: Configuring an EX4500 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 63](#)
- [Example: Expanding an EX3300 Virtual Chassis on page 67](#)
- [Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet on page 71](#)
- [Example: Adding EX4500 Switches to a Preprovisioned Virtual Chassis on page 77](#)
- [Example: Adding an EX4500 Switch to a Nonprovisioned Virtual Chassis on page 81](#)
- [Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration on page 85](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets on page 91](#)
- [Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets on page 98](#)
- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104](#)
- [Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis on page 116](#)
- [Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch on page 120](#)
- [Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch on page 126](#)
- [Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails on page 131](#)
- [Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge on page 134](#)

- [Example: Configuring Link Aggregation Groups Using EX4200 Uplink Virtual Chassis Ports on page 136](#)
- [Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches on page 144](#)

Example: Configuring an EX2200 Virtual Chassis with Four Member Switches

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 on page 50](#)
- [Overview and Topology on page 50](#)
- [Configuration on page 51](#)
- [Verification on page 52](#)
- [Troubleshooting the Virtual Chassis on page 53](#)

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

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 mastership priorities and member IDs are assigned by the user using the Junos OS software in this example. The software elects a master based on several criteria, including how long a member switch has belonged to the Virtual Chassis configuration. For additional details on how the master switch is elected, see [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#).



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 148](#).



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

The topology for this example consists of four EX2200 switches.

[Table 10 on page 51](#) shows the default configuration settings for the Virtual Chassis.

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

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

Configuration

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 *Installing a Transceiver in an EX Series Switch*.
2. Power on switch-0 (the member switch that you want to function as the master).
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 mastership priority of the switches that you want to act in the master 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 mastership priority selection 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 mastership 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

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

- [Verifying That the Member Switches are Present and the Roles are Properly Assigned on page 52](#)

[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 master 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 Mode	Neighbor ID	List Interface
0 (FPC 0)	Prsnt	GR0211195292	ex2200-24t-4g	255	Master*	NA	1	vcp-255/1/0
1 (FPC 1)	Prsnt	GP0211420824	ex2200-24t-4g	255	Backup	NA	3	vcp-255/1/1
2 (FPC 2)	Prsnt	GP0211464311	ex2200-24t-4g	128	Linecard	NA	0	vcp-255/1/0
3 (FPC 3)	Prsnt	CW0210108833	ex2200-24t-4g	128	Linecard	NA	2	vcp-255/1/1
							2	vcp-255/1/0
							3	vcp-255/1/1
							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 master, the status of each member switch, the mastership 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 master and backup roles.

Troubleshooting the Virtual Chassis

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

- [Troubleshooting the Assignment of Roles on page 53](#)
- [Troubleshooting the VCPs on page 53](#)

Troubleshooting the Assignment of Roles

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

Solution Modify the mastership priority values.

To quickly modify the mastership 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 \(CLI Procedure\) on page 148](#)

Example: Configuring an EX3300 Virtual Chassis with a Master and Backup

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 master 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 on page 54](#)
- [Overview and Topology on page 54](#)
- [Configuration on page 55](#)
- [Verification on page 56](#)
- [Troubleshooting the Virtual Chassis on page 57](#)

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

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 master 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 mastership priorities and member IDs are assigned by the software. The software elects a master based on several criteria, including how long a member switch has belonged to the Virtual Chassis configuration. For additional details, see [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#). Therefore, we recommend that you start by powering on only one member switch, the one that you want to function as the master.



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

The topology for this example consists of two EX3300 switches.

[Table 11 on page 55](#) 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	EX3300 switch	0	Master: mastership priority 128
SWA-1	EX3300 switch	1	Backup: mastership priority 128

Configuration

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

Step-by-Step Procedure

To configure a Virtual Chassis with master 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 master).
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@switch# **set no-split-detection**
7. Power on SWA-1.

Verification

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

- [Verifying That the Mastership Priority Is Assigned Appropriately on page 56](#)
- [Verifying That the VCPs Are Operational on page 56](#)

Verifying That the Mastership Priority Is Assigned Appropriately

Purpose Verify that the master, 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 master, the mastership 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 mastership 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 master. 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 or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID	Neighbor Interface
1/2	Configured	Up	10000	1	vcp-255/1/2
1/3	Configured	Up	10000	1	vcp-255/1/3

```
fpc1:
```

Interface or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID	Neighbor Interface
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

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

- [Troubleshooting the Assignment of Roles on page 57](#)
- [Troubleshooting the VCPs on page 57](#)

Troubleshooting the Assignment of Roles

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

Solution Modify the mastership priority values.

To quickly modify the mastership 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 \(CLI Procedure\) on page 152](#)

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

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 master and backup in a single wiring closet:

- [Requirements on page 58](#)
- [Overview and Topology on page 58](#)
- [Configuration on page 60](#)
- [Verification on page 60](#)
- [Troubleshooting the Virtual Chassis on page 62](#)

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. See *Mounting an EX4200 Switch*.
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

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 master 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 Switch as a Virtual Chassis Port \(CLI Procedure\)](#)” on page 188.

By default, after you interconnect the switches with the dedicated VCPs and power on the switches, the VCPs are operational. The mastership priorities and member IDs are assigned by the software. The software elects a master based on several criteria, including how long a member switch has belonged to the Virtual Chassis configuration. For additional details, see “[Understanding How the Master in a Virtual Chassis Is Elected](#)” on page 20. Therefore, we recommend that you start by powering on only one member switch, the one that you want to function as the master.



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.

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

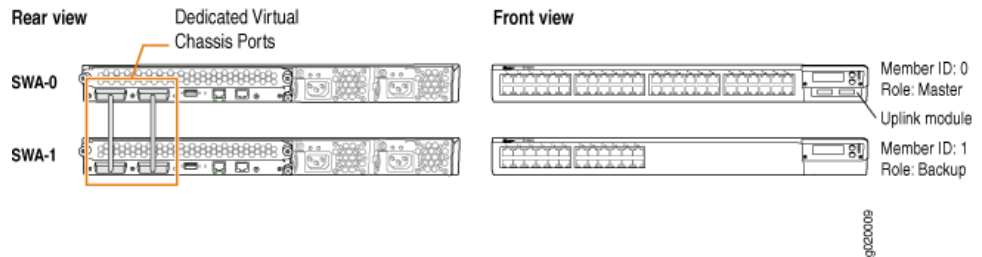
[Table 12 on page 59](#) shows the default configuration settings for the two-member Virtual Chassis.

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

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

Figure 8 on page 60 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 Master and Backup



Configuration

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

Step-by-Step Procedure

To configure a Virtual Chassis with master 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 master).
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@switch# set no-split-detection
```

7. Power on SWA-1.

Verification

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

- [Verifying That the Mastership Priority Is Assigned Appropriately on page 60](#)
- [Verifying That the VCPs Are Operational on page 61](#)

Verifying That the Mastership Priority Is Assigned Appropriately

Purpose Verify that the master, 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 ID	Neighbor 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 master, the mastership 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 mastership 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 master. 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 or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID	Neighbor Interface
vcp-0	Dedicated	Up	32000	1	vcp-1
vcp-1	Dedicated	Up	32000	1	vcp-0

```
fpc1:
```

Interface or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID	Neighbor Interface
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

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

- [Troubleshooting the Assignment of Roles on page 62](#)
- [Troubleshooting the VCPs on page 62](#)

Troubleshooting the Assignment of Roles

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

Solution Modify the mastership priority values.

To quickly modify the mastership 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 on page 71](#)
- [Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration on page 85](#)
- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)

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

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 master and backup in a single wiring closet:

- [Requirements on page 63](#)
- [Overview and Topology on page 63](#)
- [Configuration on page 64](#)
- [Verification on page 65](#)
- [Troubleshooting the Virtual Chassis on page 66](#)

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

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 master 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 mastership priorities and member IDs are assigned by the software. The software elects a master based on several criteria, including how long a member switch has belonged to the Virtual Chassis configuration. For additional details,

see [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#). Therefore, we recommend that you start by powering on only one member switch, the one that you want to function as the master.



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.

The topology for this example consists of two EX4500 switches.

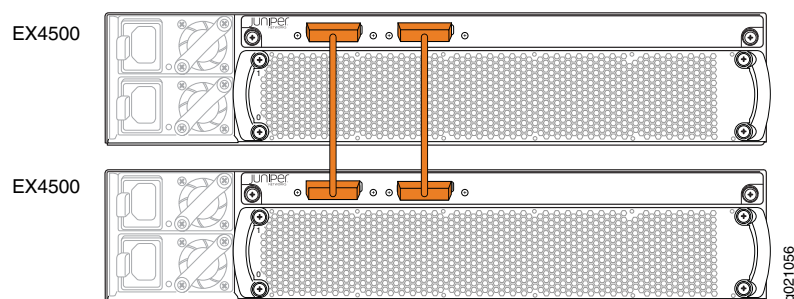
[Table 13 on page 64](#) shows the default configuration settings for the two-member Virtual Chassis.

Table 13: Components of the Basic Virtual Chassis Access Switch Topology

Member Switch	Hardware	Member ID	Role and Priority
SWA-0	EX4500 switch	0	Master: mastership priority 128
SWA-1	EX4500 switch	1	Backup: mastership priority 128

[Figure 9 on page 64](#) 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 Master and Backup



Configuration

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

Step-by-Step Procedure

To configure a Virtual Chassis with master and backup:

1. Power on SWA-0 (the member switch that you want to function as the master).
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

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

- [Verifying That the Mastership Priority Is Assigned Appropriately on page 65](#)
- [Verifying That the VCPs Are Operational on page 66](#)

Verifying That the Mastership Priority Is Assigned Appropriately

Purpose Verify that the master, 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	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 master, the mastership 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 mastership 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 master. 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 or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID	Neighbor Interface
vcp-0	Dedicated	Up	32000	1	vcp-1
vcp-1	Dedicated	Up	32000	1	vcp-0

```
fpc1:
```

Interface or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID	Neighbor Interface
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

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

- [Troubleshooting the Assignment of Roles on page 66](#)
- [Troubleshooting the VCPs on page 67](#)

Troubleshooting the Assignment of Roles

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

Solution Modify the mastership priority values.

To quickly modify the mastership 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 on page 98](#)
- [Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis on page 116](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)

Example: Expanding an EX3300 Virtual Chassis

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 on page 67](#)
- [Overview and Topology on page 67](#)
- [Configuration on page 68](#)
- [Verification on page 69](#)
- [Troubleshooting on page 71](#)

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” on page 245](#).

Overview and Topology

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 master and backup switches, explicitly configure the mastership priority of these switches, specifying the highest possible value (255) for both the master 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.

The topology for this example consists of three EX3300 switches.

[Table 14 on page 68](#) 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	EX3300 switch	0	master; mastership priority 255
SWA-1	EX3300 switch	1	backup; mastership priority 255
SWA-2	EX3300 switch	2	linecard; mastership priority 128

Configuration

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.

CLI Quick Configuration

To maintain the master 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 mastership priority of SWA-0 (member 0) to be the highest possible value, thereby ensuring that it functions as the master of the expanded Virtual Chassis configuration.

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

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

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 on page 69](#)
- [Verifying That the VCPs Are Operational on page 70](#)

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, mastership priority values, and assigned roles.

```
user@SWA-0> 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	ex3300-24t	255	Master*	1 vcp-255/1/2 2 vcp-255/1/3
1 (FPC 1)	Prsnt	def456	ex3300-24t	255	Backup	2 vcp-255/1/2 0 vcp-255/1/3
2 (FPC 2)	Prsnt	abd231	ex3300-24t	128	Linecard	0 vcp-255/1/2 1 vcp-255/1/3

Meaning The `show virtual-chassis` command lists the member switches of the Virtual Chassis configuration with the member IDs and mastership 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 mastership priority value 128. Because the mastership priority is lower than the mastership 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 mastership 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 or PIC / Port	Type	Status
1/2	Configured	Up
1/3	Configured	Up

```
fpc1:
```

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

```
fpc2:
```

Interface or PIC / Port	Type	Status
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

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

- [Troubleshooting Mastership Priority on page 71](#)
- [Troubleshooting Nonoperational VCPs on page 71](#)

Troubleshooting Mastership Priority

Problem You want to designate a different member as the master.

Solution Change the mastership priority value or values of the switches, designating the highest mastership priority value for the switch that you want to be master.

1. Lower the mastership priority of the existing master (member 0).

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

2. Set the mastership priority of the member that you want to be the master 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 \(CLI Procedure\) on page 152](#)

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

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 on page 72](#)
- [Overview and Topology on page 72](#)
- [Configuration on page 74](#)
- [Verification on page 75](#)
- [Troubleshooting on page 76](#)

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” on page 245](#).

Overview and Topology

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 91](#).

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

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 Switch as a Virtual Chassis Port \(CLI Procedure\)](#)” on page 188.

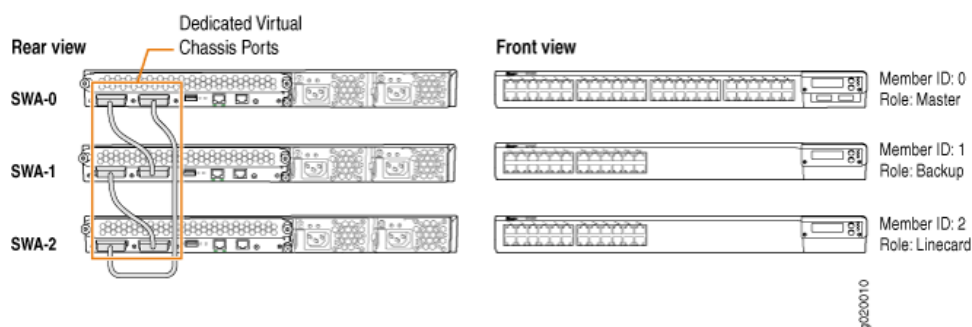
Table 15 on page 73 shows the configuration settings for the expanded Virtual Chassis.

Table 15: Components of the Expanded Virtual Chassis Access Switch

Member Switch	Hardware	Member ID	Role in Virtual Chassis
SWA-0	EX4200-48P switch	0	master; mastership priority 255
SWA-1	EX4200-24T switch	1	backup; mastership priority 255
SWA-2	EX4200-24P switch	2	linecard; mastership priority 128

Figure 10 on page 73 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

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.

CLI Quick Configuration

To maintain the master 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 mastership priority of SWA-0 (member 0) to be the highest possible value, thereby ensuring that it functions as the master of the expanded Virtual Chassis configuration.

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

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

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 on page 75](#)
- [Verifying That the VCPs Are Operational on page 76](#)

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, mastership priority values, and assigned roles.

```
user@SWA-0> 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 The `show virtual-chassis` command lists the member switches of the Virtual Chassis configuration with the member IDs and mastership 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 mastership priority value 128. Because the mastership priority is lower than the mastership 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 mastership 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 or PIC / Port	Type	Status
vcp-0	Dedicated	Up
vcp-1	Dedicated	Up

```
fpc1:
```

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

```
fpc2:
```

Interface or PIC / Port	Type	Status
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

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

- [Troubleshooting Mastership Priority on page 76](#)
- [Troubleshooting Nonoperational VCPs on page 77](#)

Troubleshooting Mastership Priority

Problem You want to designate a different member as the master.

Solution Change the mastership priority value or values of the switches, designating the highest mastership priority value for the switch that you want to be master.

1. Lower the mastership priority of the existing master (member 0).

- ```
[edit virtual-chassis]
user@SWA-0# set member 0 mastership-priority 1
```
2. Set the mastership priority of the member that you want to be the master to the highest possible value (255):

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

### Troubleshooting Nonoperational VCPs

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Problem</b>               | The VCP interface shows a status of <b>down</b> .                                                                                                                                                                                                                                                                                                                                                                |
| <b>Solution</b>              | Check the cable to make sure that it is properly and securely connected to the VCPs.                                                                                                                                                                                                                                                                                                                             |
| <b>Related Documentation</b> | <ul style="list-style-type: none"> <li>• <a href="#">Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration on page 85</a></li> <li>• <a href="#">Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) on page 156</a></li> <li>• <a href="#">Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure) on page 160</a></li> </ul> |

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

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 on page 77](#)
- [Overview and Topology on page 78](#)
- [Configuration on page 79](#)
- [Verification on page 80](#)

### 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

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.

[Table 16 on page 78](#) shows the configuration of the two-member EX4200 Virtual Chassis before the EX4500 member switches were added.

**Table 16: 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         | Master            |
| SWA-1         | EX4200 switch | 1         | Backup            |

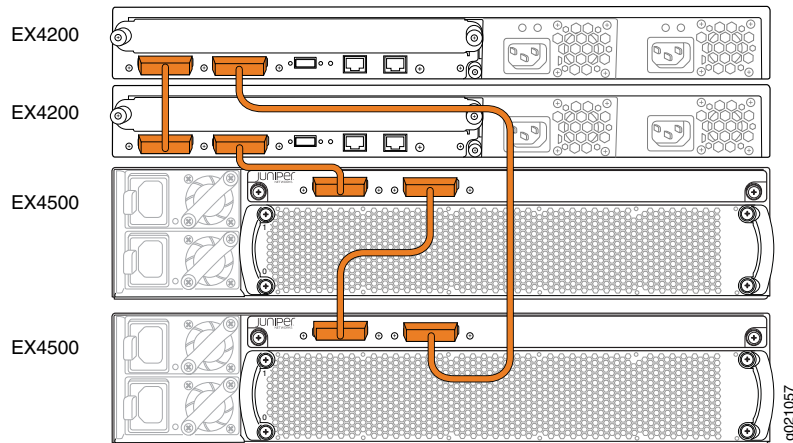
[Table 17 on page 78](#) shows the configuration of the mixed EX4200 and EX4500 Virtual Chassis after the EX4500 member switches are added to the Virtual Chassis and the roles are assigned.

**Table 17: Final Mixed EX4200 and EX4500 Virtual Chassis Components**

| Member Switch | Hardware      | Member ID | Role and Priority |
|---------------|---------------|-----------|-------------------|
| SWA-0         | EX4200 switch | 0         | Linecard          |
| SWA-1         | EX4200 switch | 1         | Linecard          |
| SWA-2         | EX4500 switch | 2         | Master            |
| SWA-3         | EX4500 switch | 3         | Backup            |

[Figure 11 on page 79](#) shows the hardware topology of the mixed EX4200 and EX4500 Virtual Chassis.

Figure 11: Mixed EX4200 and EX4500 Virtual Chassis Topology (Preprovisioned Configuration)



## Configuration

### 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

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

- [Verifying Virtual Chassis Availability and Roles on page 80](#)

### 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     |
| 1 (FPC 1) | Prsnt  | AK02073602814 | ex4200-48p | 0                   | Linecard | 3                | vcp-1     |
|           |        |               |            |                     |          | 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 master, the mastership 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 mastership priorities and roles.

#### Related Documentation

- [Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets on page 98](#)
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 178](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)

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

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 on page 81](#)
- [Overview and Topology on page 82](#)
- [Configuration on page 83](#)
- [Verification on page 85](#)

#### 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

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 master election algorithm that determines the Virtual Chassis roles for each member switch. The master election algorithm first checks the mastership priority ID. The mastership priority ID is any number between 0 and 255. The switch with the higher mastership priority ID is elected into the master role and the other switch, unless it has been configured with a mastership priority value of 0, is elected into the backup role.

The default mastership priority ID for an EX4200 or EX4500 switch is 128. Both EX4200 switches are using this default mastership 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.

Table 18 on page 82 shows the default configuration settings for the two-member Virtual Chassis before the EX4500 member switches are added to the Virtual Chassis.

**Table 18: 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         | Master: mastership priority 128 |
| SWA-1         | EX4200 switch | 1         | Backup: mastership priority 128 |

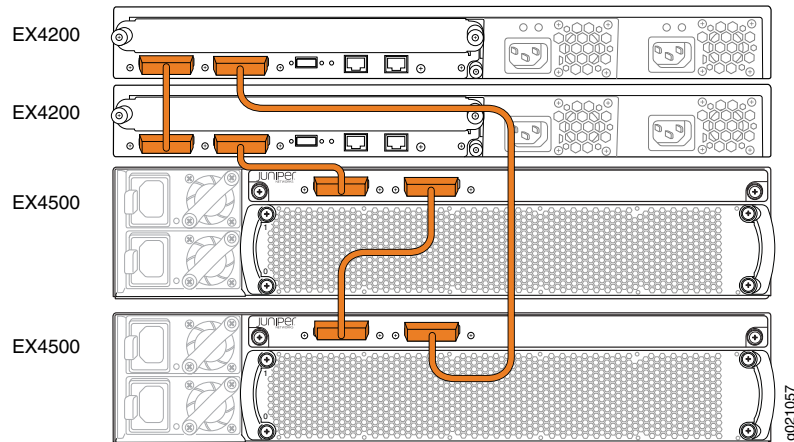
Table 19 on page 82 shows the configuration of the mixed EX4200 and EX4500 Virtual Chassis after the EX4500 member switches are added to the Virtual Chassis and the EX4500 switches have been configured in the master and backup roles.

**Table 19: Final Mixed EX4200 and EX4500 Virtual Chassis Components**

| Member Switch | Hardware      | Member ID | Role and Priority               |
|---------------|---------------|-----------|---------------------------------|
| SWA-0         | EX4200 switch | 0         | Linecard: mastership priority 0 |
| SWA-1         | EX4200 switch | 1         | Linecard: mastership priority 0 |
| SWA-2         | EX4500 switch | 2         | Master: mastership priority 255 |
| SWA-3         | EX4500 switch | 3         | Backup: mastership priority 255 |

Figure 12 on page 83 shows the hardware topology of the mixed EX4200 and EX4500 Virtual Chassis.

**Figure 12: Mixed EX4200 and EX4500 Virtual Chassis Topology (Nonprovisioned Configuration)**



## Configuration

### 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 master 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 mastership 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 mastership priority ID of 0 never assumes the master 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 master or backup roles, assign the switches a mastership priority between 1 and 255.



## Verification

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

- [Verifying Virtual Chassis Availability and Roles on page 85](#)

### 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 ID | List Interface |
|-----------|--------|---------------|------------|---------------------|----------|-------------|----------------|
| 0 (FPC 0) | Prsnt  | AK02073602764 | ex4200-48p | 0                   | Linecard | 1           | vcp-0          |
| 1 (FPC 1) | Prsnt  | AK02073602814 | ex4200-48p | 0                   | Linecard | 3           | vcp-1          |
| 2 (FPC 2) | Prsnt  | AK02073602844 | ex4500-40f | 255                 | Master*  | 0           | vcp-0          |
| 3 (FPC 3) | Prsnt  | AK02073602088 | ex4500-40f | 255                 | Backup   | 2           | vcp-1          |
|           |        |               |            |                     |          | 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 master, the mastership 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 mastership priorities and roles.

- Related Documentation**
- [Example: Connecting EX4500 Member Switches in a Virtual Chassis Across Wiring Closets on page 98](#)
  - [Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 180](#)
  - [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)

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

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 master, backup, and linecard roles, based on the sequence in which the switches are powered on and other factors in the master election algorithm.



**TIP:** We recommend that you explicitly configure the mastership 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 on page 86](#)
- [Overview and Topology on page 86](#)
- [Configuration on page 87](#)
- [Verification on page 88](#)
- [Troubleshooting on page 90](#)

## 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

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.

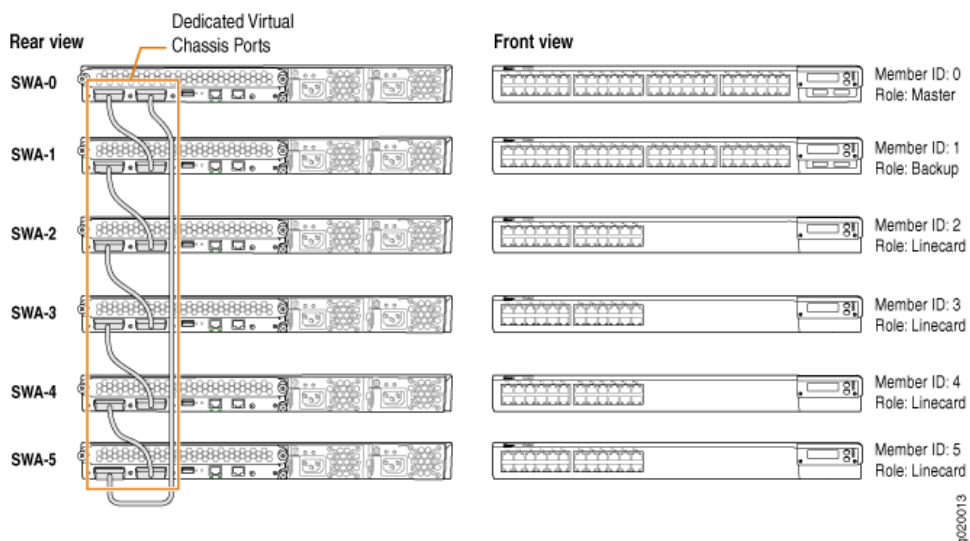
To set up a multimember Virtual Chassis configuration within a single wiring closet, you need to run the EZSetup program only once. Connect to the master and run EZSetup to specify its identification, time zone, and network properties. When additional switches are connected through the Virtual Chassis ports (VCPs), they automatically receive the same properties that were specified for the master.

The topology for this example (see [Figure 13 on page 87](#)) 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 87 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

To configure a multimember Virtual Chassis access switch in a single wiring closet using the factory defaults, perform this task:

### 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 master (SWA-0). This example uses one of the larger switches (EX4200-48P) as the master.
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 master, 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

To confirm that the configuration is working properly, perform these tasks:

- [Verifying the Member IDs and Roles of the Member Switches on page 88](#)
- [Verifying That the VCPs Are Operational on page 88](#)

### 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<br>Priority | Role     | Neighbor List<br>ID Interface |
|-----------|--------|-----------|------------|------------------------|----------|-------------------------------|
| 0 (FPC 0) | Prsnt  | abc123    | ex4200-48p | 128                    | Master*  | 1 vcp-0<br>5 vcp-1            |
| 1 (FPC 1) | Prsnt  | def123    | ex4200-48p | 128                    | Backup   | 2 vcp-0<br>0 vcp-1            |
| 2 (FPC 2) | Prsnt  | abd231    | ex4200-24p | 128                    | Linecard | 3 vcp-0<br>1 vcp-1            |
| 3 (FPC 3) | Prsnt  | cab123    | ex4200-24p | 128                    | Linecard | 4 vcp-0<br>2 vcp-1            |
| 4 (FPC 4) | Prsnt  | fed456    | ex4200-24p | 128                    | Linecard | 5 vcp-0<br>3 vcp-1            |
| 5 (FPC 5) | Prsnt  | jkl231    | ex4200-24p | 128                    | Linecard | 0 vcp-0<br>4 vcp-1            |

**Meaning** The `show virtual-chassis` command lists the member switches of the Virtual Chassis configuration with the member IDs and mastership 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<br>or<br>PIC / Port | Type      | Status |
|-------------------------------|-----------|--------|
| vcp-0                         | Dedicated | Up     |
| vcp-1                         | Dedicated | Up     |

```
fpc1:
```

| Interface<br>or<br>PIC / Port | Type      | Status |
|-------------------------------|-----------|--------|
| vcp-0                         | Dedicated | Up     |
| vcp-1                         | Dedicated | Up     |

```
fpc2:
```

| Interface<br>or<br>PIC / Port | Type      | Status |
|-------------------------------|-----------|--------|
| vcp-0                         | Dedicated | Up     |
| vcp-1                         | Dedicated | Up     |

```
fpc3:
```

| Interface<br>or<br>PIC / Port | Type      | Status |
|-------------------------------|-----------|--------|
| vcp-0                         | Dedicated | Up     |
| vcp-1                         | Dedicated | Up     |

```
fpc4:
```

| Interface<br>or<br>PIC / Port | Type      | Status |
|-------------------------------|-----------|--------|
| vcp-0                         | Dedicated | Up     |
| vcp-1                         | Dedicated | Up     |

```
fpc5:
```

| Interface<br>or<br>PIC / Port | Type      | Status |
|-------------------------------|-----------|--------|
| 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

To troubleshoot the configuration of a multimember Virtual Chassis in a single wiring closet, perform these tasks:

- [Troubleshooting Mastership Priority on page 90](#)
- [Troubleshooting Nonoperational VCPs on page 90](#)

---

### Troubleshooting Mastership Priority

---

**Problem** You want to explicitly designate one member as the master and another as backup.

**Solution** Change the mastership priority value of the member that you want to function as master, assigning the highest mastership priority value to that member.



**NOTE:** These configuration changes are made through the current master, SWA-0.

- 
1. Configure mastership 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 mastership 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 Master in a Virtual Chassis Is Elected on page 20](#)
- [Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets on page 91](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)

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

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 136.



**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 on page 91](#)
- [Overview and Topology on page 92](#)
- [Configuration on page 94](#)
- [Verification on page 96](#)
- [Troubleshooting on page 98](#)

### 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 master role, SWA-0, and run the EZSetup program (see [Table 20 on page 93](#) 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 195](#).
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 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.

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 mastership priority value (255) for SWA-0 to make it the master before we power on SWA-1. Because SWA-0 and SWA-1 are interconnected with



the dedicated VCPs, the master 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 master of the Virtual Chassis configuration, the master 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 mastership priority value for SWA-2 (255) that we configured for the master. Because we power on SWA-0 before we power on SWA-2, SWA-0 has additional prioritization properties that allow it to retain mastership of the Virtual Chassis configuration. We recommend setting identical mastership priority values for the master and backup members for high availability and smooth transition of mastership in case the original master becomes unavailable. (Setting identical mastership priority values for the master and backup members prevents the previous master from pre-empting the master role from the new master when the previous master 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 master 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 master and interconnect that uplink VCP with an uplink VCP on SWA-1.

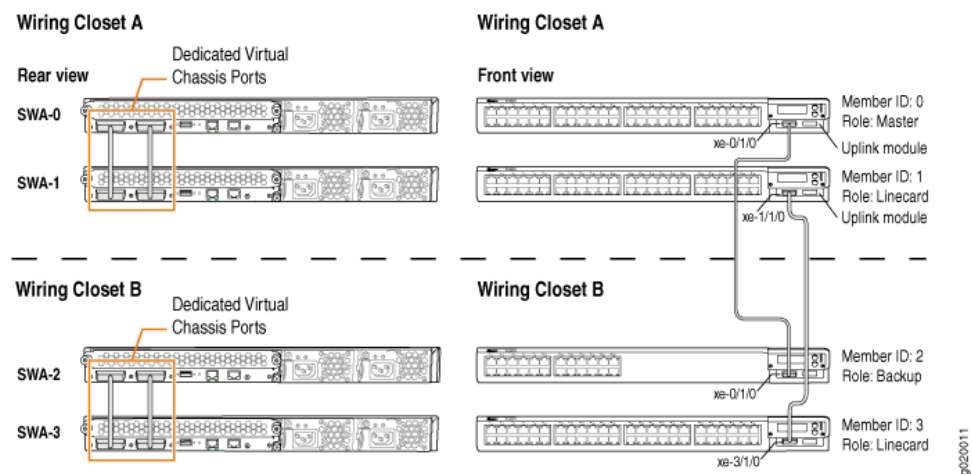
[Table 20 on page 93](#) shows the Virtual Chassis configuration settings for a Virtual Chassis composed of member switches in different wiring closets.

**Table 20: Components of a Virtual Chassis Interconnected Across Multiple Wiring Closets**

| Switch | Member ID | Role and Priority                 | Location        |
|--------|-----------|-----------------------------------|-----------------|
| SWA-0  | 0         | master; mastership priority 255   | Wiring closet A |
| SWA-1  | 1         | linecard; mastership priority 128 | Wiring closet A |
| SWA-2  | 2         | backup; mastership priority 255   | Wiring closet B |
| SWA-3  | 3         | linecard; mastership priority 128 | Wiring closet B |

[Figure 14 on page 94](#) 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

To configure the Virtual Chassis across multiple wiring closets, perform this task:

### Step-by-Step Procedure

To configure a Virtual Chassis across multiple wiring closets:

1. Configure the mastership priority of SWA-0 (member 0) to be the highest possible value (255), thereby ensuring that it functions as the master 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 mastership 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 mastership 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 master 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 mastership priority value (255), SWA-0 was powered on first and thus has longer uptime. This results in SWA-0 retaining mastership 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 master 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 mastership 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

To confirm that the configuration is working properly, perform these tasks:

- [Verifying the Member IDs and Roles of the Member Switches on page 96](#)
- [Verifying that the Dedicated VCPs and Uplink VCPs Are Operational on page 97](#)

### 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 Priority | Role     | Neighbor List ID Interface          |
|-----------|--------|-----------|------------|---------------------|----------|-------------------------------------|
| 0 (FPC 0) | Prsnt  | abc123    | ex4200-48p | 255                 | Master*  | 1 vcp-0<br>1 vcp-1<br>2 vcp-255/1/0 |
| 1 (FPC 1) | Prsnt  | def456    | ex4200-24t | 128                 | Linecard | 0 vcp-0<br>0 vcp-1<br>3 vcp-255/1/0 |
| 2 (FPC 2) | Prsnt  | ghi789    | ex4200-48p | 255                 | Backup   | 3 vcp-0<br>3 vcp-1<br>0 vcp-255/1/0 |
| 3 (FPC 3) | Prsnt  | jk1012    | ex4200-24t | 128                 | Linecard | 2 vcp-0<br>2 vcp-1<br>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 master, the mastership 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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>ID | Interface   |
|-------------------------------|------------|-------------|--------|-----------------|----------------|-------------|
| 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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>ID | Interface   |
|-------------------------------|------------|-------------|--------|-----------------|----------------|-------------|
| 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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>ID | Interface   |
|-------------------------------|------------|-------------|--------|-----------------|----------------|-------------|
| 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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>ID | Interface   |
|-------------------------------|------------|-------------|--------|-----------------|----------------|-------------|
| 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

To troubleshoot a Virtual Chassis configuration that is interconnected across wiring closets, perform these tasks:

- [Troubleshooting Nonoperational VCPs on page 98](#)

---

### Troubleshooting Nonoperational VCPs

---

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Problem</b>               | An uplink VCP shows a status of <b>down</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Solution</b>              | <ul style="list-style-type: none"><li>• Check the cable to make sure that it is properly and securely connected to the ports.</li><li>• If the VCP is an uplink module port, make sure that it has been explicitly set as an uplink VCP.</li><li>• If the VCP is an uplink module port, make sure that you have specified the options (<i>pic-slot</i>, <i>port</i>, and <i>member</i>) correctly.</li></ul>                                                                                                                                                      |
| <b>Related Documentation</b> | <ul style="list-style-type: none"><li>• <a href="#">Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58</a></li><li>• <a href="#">Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet on page 71</a></li><li>• <a href="#">Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration on page 85</a></li><li>• <a href="#">Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port (CLI Procedure) on page 188</a></li></ul> |

---

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

---

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 on page 99](#)
- [Overview and Topology on page 99](#)
- [Configuration on page 101](#)
- [Verification on page 103](#)
- [Troubleshooting on page 104](#)

## 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 165](#) 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 195](#).
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 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.

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 21 on page 100](#) shows the Virtual Chassis configuration settings for a Virtual Chassis composed of member switches in different wiring closets.

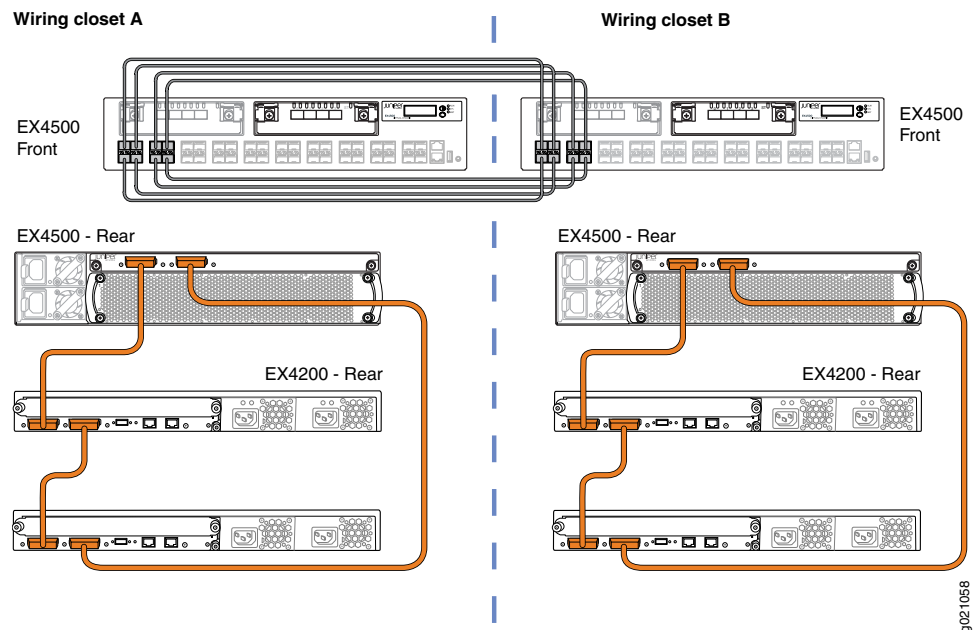
**Table 21: Components of a Virtual Chassis Interconnected Across Wiring Closets**

| Switch | Model         | Member ID | Role     | Location        |
|--------|---------------|-----------|----------|-----------------|
| SWA-0  | EX4500 switch | 0         | Master   | 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 101](#) 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

To configure the Virtual Chassis across wiring closets, perform this task:

### 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:
4. Power on the remaining switches.
5. Configure all switches individually as members of the mixed EX4200 and EX4500 Virtual Chassis:
6. Reboot all switches:
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

To confirm that the configuration is working properly, perform these tasks:

- [Verifying the Member IDs and Roles of the Member Switches on page 103](#)

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

Member ID	Status	Serial No	Model	Mastership 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 master and the roles. It also displays the neighbor members with which each member is interconnected.

Troubleshooting

To troubleshoot a Virtual Chassis configuration that is interconnected across wiring closets, perform these tasks:

- [Troubleshooting Nonoperational VCPs on page 104](#)

[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 \(CLI Procedure\) on page 193](#)

Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File

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 mastership 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 \(CLI Procedure\)”](#) on page 171.

This example describes how to configure a Virtual Chassis across multiple wiring closets using a preprovisioned configuration file:

- [Requirements on page 105](#)
- [Overview and Topology on page 106](#)
- [Configuration on page 110](#)
- [Verification on page 113](#)
- [Troubleshooting on page 115](#)

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 master or backup. If you configure the member with a **line-card** role, it is not eligible to become a master 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 master 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 master 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

You must select two members that you want to make eligible for election as master 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 master of the Virtual Chassis configuration and the other will function as the backup.

You designate additional members, which are not eligible for election as master, 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 master 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 master 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 master 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 master of the Virtual Chassis configuration, the master 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 master 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 master can now detect that all members are present. Finally, for redundancy, configure an additional VCP on SWA-7 through the master.

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 22 on page 108](#) shows the Virtual Chassis configuration settings for a preprovisioned Virtual Chassis composed of member switches in different wiring closets.

Table 22: 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	xe-0/1/0	EX4200-48P and XFP uplink module	Wiring closet A
SWA-1	def456	1	linecard		EX4200-24T	Wiring closet A
SWA-2	ghi789	2	linecard	xe-2/1/0	EX4200-48P and XFP uplink module	Wiring closet A
SWA-3	jkl012	3	linecard		EX4200-24T	Wiring closet A
SWA-4	mno345	4	linecard		EX4200-48P	Wiring closet A

Table 22: Components of a Preprovisioned EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets (*continued*)

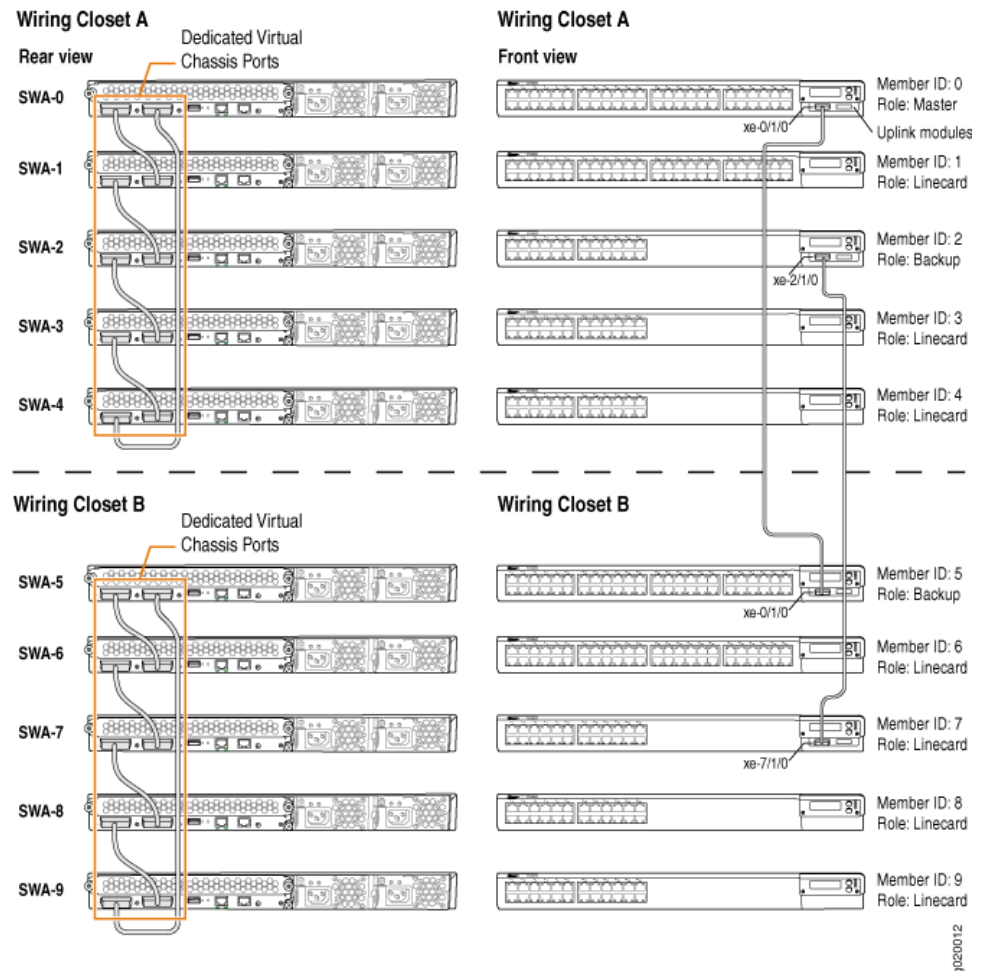
Switch	Serial number	Member ID	Role	Uplink Module Ports	Hardware	Location
SWA-5	pqr678	5	routing-engine	xe-0/1/0 <small>NOTE: The member ID of SWA-5 is 0 at the time that its uplink module port is configured as a VCP.</small>	EX4200-48P and XFP uplink module	Wiring closet B
SWA-6	stu901	6	linecard		EX4200-24T	Wiring closet B
SWA-7	vwx234	7	linecard	xe-7/1/0	EX4200-24T and XFP uplink module	Wiring closet B
SWA-8	yla567	8	linecard		EX4200-24T	Wiring closet B
SWA-9	bcd890	9	linecard		EX4200-48P	Wiring closet B

Figure 16 on page 110 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 110 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

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.

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 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 master 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 master 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 master 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 master 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

To confirm that the configuration is working properly, perform these tasks:

- [Verifying the Member IDs and Roles of the Member Switches on page 113](#)
- [Verifying That the Dedicated VCPs and Uplink VCPs Are Operational on page 114](#)

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 Priority	Role	Neighbor List 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	jk1012	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	vwx234	ex4200-24t	0	Linecard	8 vcp-0 6 vcp-1 2 1/0
8 (FPC 8)	Prsnt	yza567	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 master 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 or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0	Dedicated	Up			
vcp-1	Dedicated	Up			
1/0	Configured	Up			

```
fpc1:
```

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

```
fpc2:
```

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

```
fpc3:
```

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

```
fpc4:
```

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

fpc5:

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

fpc6:

Interface or PIC / Port	Type	Status	Speed (mbps)	Neighbor ID Interface
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

To troubleshoot a preprovisioned Virtual Chassis configuration that is interconnected across wiring closets, perform these tasks:

- [Troubleshooting Nonoperational VCPs on page 116](#)

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 Master and Backup in a Single Wiring Closet on page 58](#)
 - [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets on page 91](#)
 - [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
 - [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)

Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis

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 mastership 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 on page 116](#)
- [Overview and Topology on page 117](#)
- [Configuration on page 118](#)
- [Verification on page 119](#)
- [Troubleshooting on page 120](#)

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

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 master, 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 master 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 master 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.

The topology for this example consists of:

- Two EX4500 switches (SWA-0 and SWA-1)
- Two EX4200 switches (SWA-2 and SWA-3)

Table 23 on page 118 shows the Virtual Chassis configuration settings for a preprovisioned Virtual Chassis.

Table 23: 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

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.

Step-by-Step Procedure

To create a preprovisioned configuration for the Virtual Chassis:

1. Power on the EX4500 switch (SWA-0) in the master 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

To confirm that the configuration is working properly, perform these tasks:

- [Verifying the Member IDs and Roles of the Member Switches on page 119](#)

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 Priority	Role	Neighbor List 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	jk1012	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 master 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

To troubleshoot a preprovisioned Virtual Chassis configuration that is interconnected across wiring closets, perform these tasks:

- [Troubleshooting Nonoperational VCPs on page 120](#)

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 \(CLI Procedure\) on page 176](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)

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

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. See *Understanding Aggregated Ethernet Interfaces and LACP* for more information.

This example describes how to configure uplink LAGs to connect a Virtual Chassis access switch to a Virtual Chassis distribution switch:

- [Requirements on page 121](#)
- [Overview and Topology on page 121](#)
- [Configuration on page 123](#)

- [Verification on page 125](#)
- [Troubleshooting on page 126](#)

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 156](#).
- 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

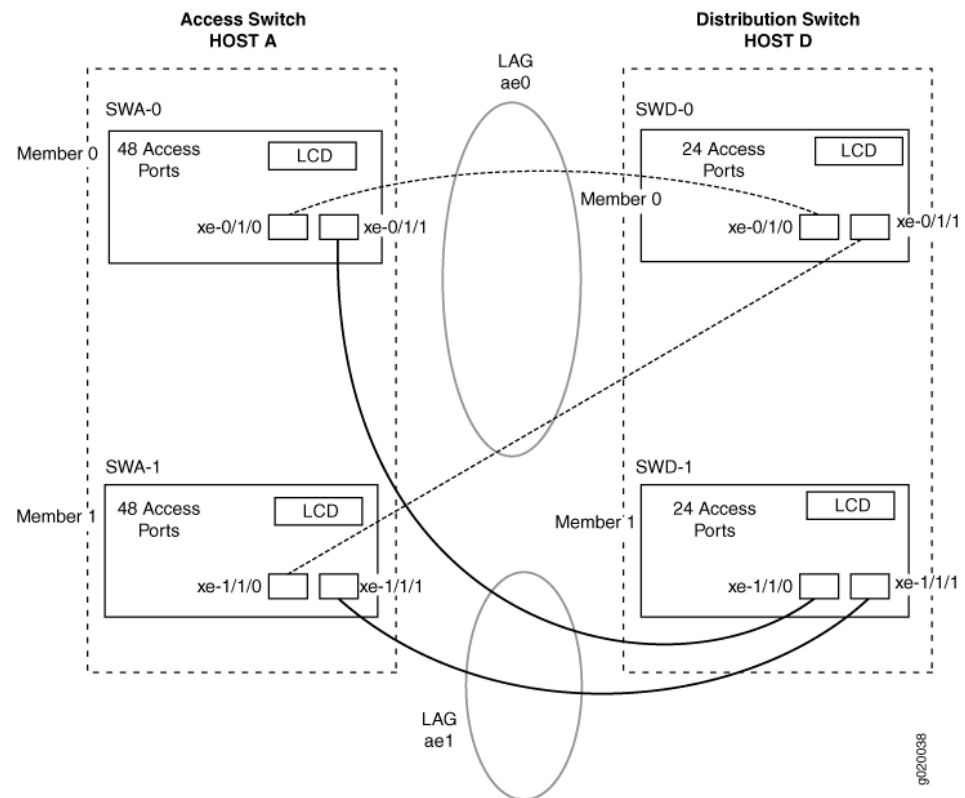


Table 24 on page 123 details the topology used in this configuration example.

Table 24: 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

To configure two uplink LAGs from the Virtual Chassis access switch to the Virtual Chassis distribution switch:

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:

- Specify the number of LAGs to be created on the chassis:


```
[edit chassis]
user@Host-A# set aggregated-devices ethernet device-count 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

To verify that switching is operational and two LAGs have been created, perform these tasks:

- [Verifying That LAG ae0 Has Been Created on page 125](#)
- [Verifying That LAG ae1 Has Been Created on page 125](#)

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

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 Master and Backup in a Single Wiring Closet on page 58](#)
- [Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch on page 126](#)
- *Example: Connecting an Access Switch to a Distribution Switch.*
- *Virtual Chassis Cabling Configuration Examples for EX4200 Switches*
- *Installing an Uplink Module in an EX4200 Switch*
- *Uplink Modules in EX4200 Switches*

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

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 120:](#)

- [Requirements on page 127](#)
- [Overview and Topology on page 127](#)
- [Configuring LACP for the LAGs on the Virtual Chassis Access Switch on page 128](#)
- [Configuring LACP for the LAGs on the Virtual Chassis Distribution Switch on page 128](#)
- [Verification on page 129](#)
- [Troubleshooting on page 130](#)

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 156](#).
- 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 120](#).

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 120](#). 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

To configure LACP for the access switch LAGs, perform these tasks:

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

To configure LACP for the two uplink LAGs from the Virtual Chassis access switch to the Virtual Chassis distribution switch, perform these tasks:

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

To verify that LACP packets are being exchanged, perform these tasks:

- [Verifying the LACP Settings on page 129](#)
- [Verifying That the LACP Packets Are Being Exchanged on page 130](#)

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

To troubleshoot a nonworking LACP link, perform these tasks:

- [Troubleshooting a Nonworking LACP Link on page 130](#)

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 Access Switch to a Distribution Switch*
- *Virtual Chassis Cabling Configuration Examples for EX4200 Switches*
- *Installing an Uplink Module in an EX4200 Switch*
- *Understanding Aggregated Ethernet Interfaces and LACP*

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

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 on page 131](#)
- [Overview and Topology on page 131](#)
- [Configuration on page 133](#)
- [Verification on page 133](#)

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 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

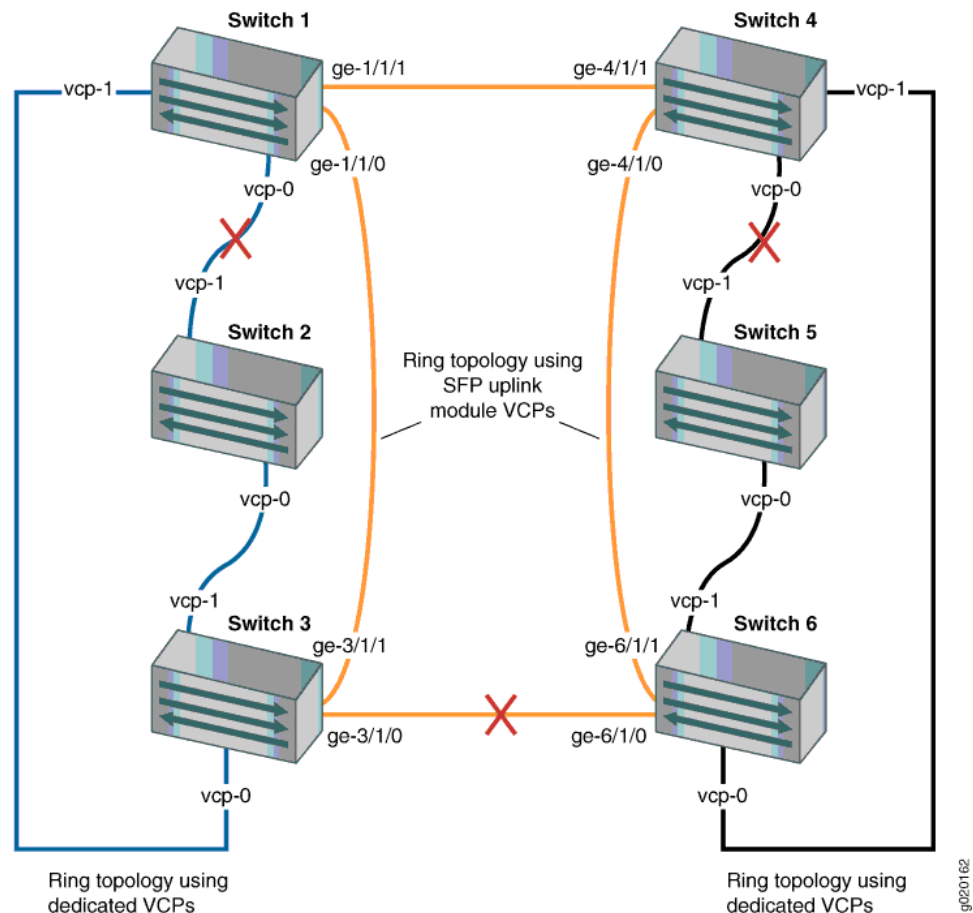
is enabled on all dedicated VCPs. If you configure uplink module ports as VCPs, you must manually configure fast failover on these ports.

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 132 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



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

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
```

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

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 on page 133](#)

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 on page 197](#)
 - [Disabling Fast Failover in an EX Series Virtual Chassis on page 198](#)
 - [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
 - [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)

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

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 on page 134](#)
- [Overview and Topology on page 135](#)
- [Configuration on page 135](#)
- [Verification on page 135](#)

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

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.

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 master is switch-A.

Configuration

To assign the Virtual Chassis ID in a Virtual Chassis configuration:

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

To verify that the Virtual Chassis ID has been assigned as you intended, perform these tasks:

- [Verifying That the Virtual Chassis ID Is Assigned on page 136](#)

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	<ul style="list-style-type: none">• Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure) on page 201• Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) on page 156• Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure) on page 160

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

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 EX Series Virtual Chassis Port Link Aggregation” on page 26](#).

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 on page 137](#)
- [Overview and Topology on page 137](#)
- [Configuration on page 139](#)
- [Verification on page 141](#)
- [Troubleshooting on page 144](#)

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 195](#).
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 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.

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 mastership priority value (255) for SWA-0 to make it the master before we power on SWA-1. Because SWA-0 and SWA-1 are interconnected with the dedicated VCPs, the master 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 mastership priority value for SWA-2 (255) that we configured for the master. Because we power on SWA-0 before we power on SWA-2, SWA-0 retains mastership of the Virtual Chassis configuration.

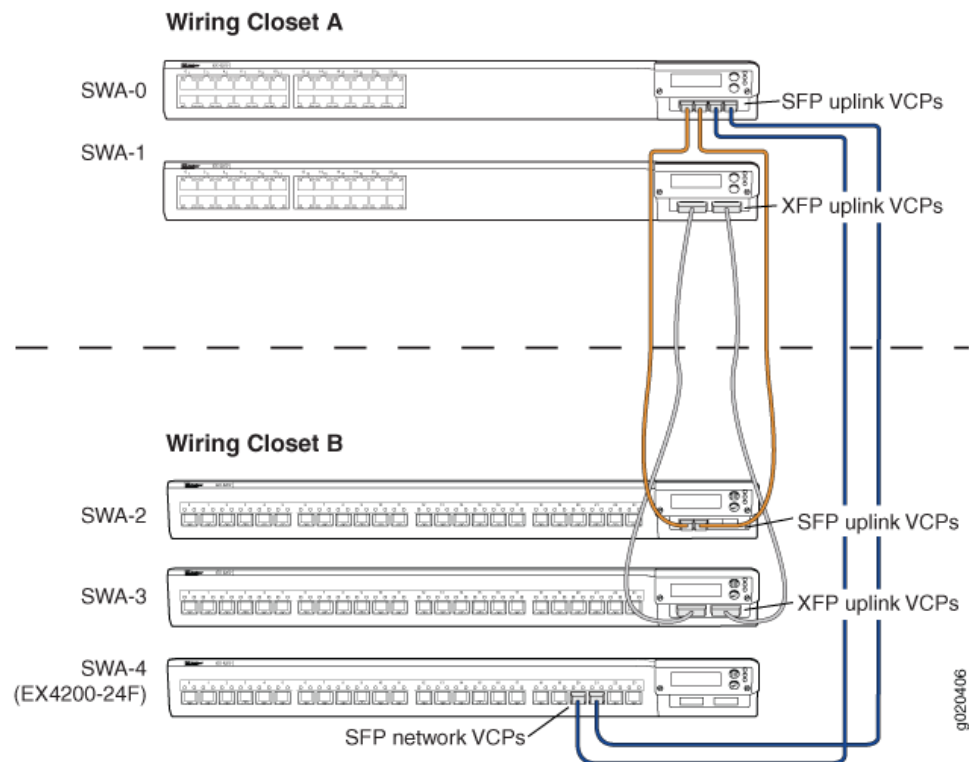


NOTE: We recommend setting identical mastership priority values for the master and backup members for high availability and smooth transition of mastership in case the original master becomes unavailable.

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 138 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

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:

Step-by-Step Procedure

To configure a Virtual Chassis across multiple wiring closets and interconnect them to form LAGs:

1. Configure the mastership priority of SWA-0 (member 0) to be the highest possible value (255), thereby ensuring that it functions as the master 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 mastership 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 mastership 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 master 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 mastership priority value (255), SWA-0 was powered on first and thus has longer uptime. This results in SWA-0 retaining mastership 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 mastership 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

To confirm that the configuration is working properly, perform these tasks:

- [Verifying the Member IDs and Roles of the Member Switches on page 141](#)
- [Verifying That the VCPs Are Operational on page 142](#)

### 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<br>Priority | Role    | Neighbor List<br>ID Interface                                                            |
|-----------|--------|-----------|------------|------------------------|---------|------------------------------------------------------------------------------------------|
| 0 (FPC 0) | Prsnt  | abc123    | ex4200-48p | 255                    | Master* | 1 vcp-0<br>1 vcp-1<br>2 vcp-255/1/0<br>2 vcp-255/1/1<br>4 vcp-255/0/20<br>4 vcp-255/0/21 |

|           |       |        |            |     |          |                                                      |
|-----------|-------|--------|------------|-----|----------|------------------------------------------------------|
| 1 (FPC 1) | Prsnt | def456 | ex4200-24t | 128 | Linecard | 0 vcp-0<br>0 vcp-1<br>3 vcp-255/1/0<br>3 vcp-255/1/1 |
| 2 (FPC 2) | Prsnt | ghi789 | ex4200-48p | 255 | Backup   | 3 vcp-0<br>3 vcp-1<br>0 vcp-255/1/0<br>0 vcp-255/1/1 |
| 3 (FPC 3) | Prsnt | jkl012 | ex4200-24t | 128 | Linecard | 2 vcp-0<br>2 vcp-1<br>1 vcp-255/1/0<br>1 vcp-255/1/1 |
| 4 (FPC 4) | Prsnt | mno345 | ex4200-24f | 128 | Linecard | 0 vcp-255/1/2<br>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 master, the mastership 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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>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<br>or<br>PIC / Port | Type       | Trunk<br>ID | Status | Speed<br>(mbps) | Neighbor<br>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 138](#), 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

To troubleshoot a Virtual Chassis configuration that is interconnected across wiring closets, perform this task:

- [Troubleshooting Nonoperational VCPs on page 144](#)

---

### 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 Master and Backup in a Single Wiring Closet on page 58](#)
- [Example: Expanding an EX4200 Virtual Chassis in a Single Wiring Closet on page 71](#)
- [Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration on page 85](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 188](#)

---

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

---

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 master, the master 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 on page 145](#)
- [Overview and Topology on page 145](#)
- [Configuration on page 145](#)
- [Verification on page 146](#)

## 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” on page 40](#).

## Overview and 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 master. If the software version on the new switch is not the same as the version running on the master, the master keeps the new switch in the inactive state.

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 master 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 master, 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

To configure automatic software update, perform this task:

### 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;
}
```

Verification

To verify that the software version on the new switch has been updated and that the switch has joined the Virtual Chassis configuration, perform this task:

- [Verifying That the Software Version Is Updated on page 146](#)

Verifying That the Software Version Is Updated

Purpose Verify that the new switch has joined the Virtual Chassis configuration.



NOTE: If the software version on the new switch had not been updated successfully, the master would not allow the switch to join the Virtual Chassis configuration.

Action Issue the `show virtual-chassis` command.

```
user@switch> 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	ex4200-24t	255	Master*	1	vcp-1
1 (FPC 1)	Prsnt	AK0207360281	ex4200-24t	255	Backup	2	vcp-1
2 (FPC 2)	Prsnt	AJ0207391130	ex4200-48p	128	Linecard	0	vcp-1
						1	vcp-0

Meaning Because in the initial two-member Virtual Chassis configuration member 0 was the master 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 \(CLI Procedure\) on page 199](#)
 - [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\) on page 171](#)

CHAPTER 3

Configuration Tasks

- [Configuring an EX2200 Virtual Chassis \(CLI Procedure\) on page 148](#)
- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) on page 152](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
- [Installing Software on a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Switches \(CLI Procedure\) on page 168](#)
- [Adding a New Switch to an Existing EX3300 Virtual Chassis \(CLI Procedure\) on page 169](#)
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\) on page 171](#)
- [Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) on page 176](#)
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 178](#)
- [Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 180](#)
- [Replacing a Member Switch of a Virtual Chassis Configuration \(CLI Procedure\) on page 181](#)
- [Removing an EX4200, EX4500, or EX4550 Switch From a Mixed Virtual Chassis \(CLI Procedure\) on page 184](#)
- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 185](#)
- [Setting a Port on an EX2200 Switch as a Virtual Chassis Port \(CLI Procedure\) on page 187](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 188](#)
- [Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) on page 193](#)
- [Setting an Uplink Port as a Virtual Chassis Port Using the LCD Panel on page 194](#)
- [Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\) on page 195](#)

- [Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis \(CLI Procedure\) on page 196](#)
- [Configuring Fast Failover in an EX Series Virtual Chassis on page 197](#)
- [Disabling Fast Failover in an EX Series Virtual Chassis on page 198](#)
- [Disabling Split and Merge in a Virtual Chassis \(CLI Procedure\) on page 199](#)
- [Configuring Automatic Software Update on Virtual Chassis Member Switches \(CLI Procedure\) on page 199](#)
- [Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge \(CLI Procedure\) on page 201](#)
- [Configuring Graceful Routing Engine Switchover in a Virtual Chassis \(CLI Procedure\) on page 201](#)
- [Resetting the EX4500 Switch to Run as a Standalone Switch \(CLI Procedure\) on page 202](#)
- [Enabling VLAN Pruning for Broadcast, Multicast, and Unknown Unicast Traffic in an EX Series Virtual Chassis \(CLI Procedure\) on page 203](#)

Configuring an EX2200 Virtual Chassis (CLI Procedure)

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 master sequentially assigns a member ID to other member switches. The role is determined by the mastership priority value and other factors in the master election algorithm.



NOTE: A Virtual Chassis configuration has two Routing Engines—the master switch and the backup switch. Therefore, we recommend that you always use `commit synchronize` rather than simply `commit` to save configuration changes made for a Virtual Chassis. This ensures that the configuration changes are saved 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 master 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 on page 149](#)
- [Configuring an EX2200 Virtual Chassis with a Nonprovisioned Configuration File on page 150](#)

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 master or backup role. If you configure the member with a **line-card** role, it is not eligible to function in the master or backup role.
3. Power on only the switch that you plan to use as the master switch.
4. Run the EZSetup program on the master 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 master switch apply to the entire Virtual Chassis configuration.

5. (Optional) Configure the master 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 EX Series Virtual Chassis Port Link Aggregation” on page 26](#).



NOTE: You cannot modify the mastership priority when you are using a preprovisioned configuration. The mastership priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file. The two Routing Engines are assigned the same mastership priority value. However, the member that was powered on first has higher prioritization according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#).

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 master switch.
2. Run the EZSetup program on the master 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 master switch apply to the entire Virtual Chassis configuration.

3. (Optional) Configure the master 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 mastership 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 master 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 EX Series Virtual Chassis Port Link Aggregation” on page 26](#).



NOTE: If you do not edit the Virtual Chassis configuration file, a nonprovisioned configuration is generated by default. The mastership priority value for each member switch is 128. The master role is selected by default. You can change the role that is performed by the members by modifying the mastership priority. See [“Configuring Mastership of a Virtual Chassis \(CLI Procedure\)” on page 185](#). We recommend that you specify the same mastership priority value for the desired master and backup members. In this example, the highest possible mastership priority has been assigned to two members. However, the member that was powered on first has higher prioritization according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#). The other members use the default mastership priority in this example, which configures them to function in the role of linecard.



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

Related Documentation

- [Understanding EX2200 Virtual Chassis on page 8](#)
- [Understanding EX2200 Virtual Chassis Hardware Configuration](#)
- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 185](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)

Configuring an EX3300 Virtual Chassis (CLI Procedure)

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

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 master sequentially assigns a member ID to other member switches. The role is determined by the mastership priority value and other factors in the master election algorithm.

This topic includes:

- [Configuring an EX3300 Virtual Chassis with a Preprovisioned Configuration File on page 153](#)
- [Configuring an EX3300 Virtual Chassis with a Nonprovisioned Configuration File on page 154](#)

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 master or backup role. If you configure the member with a **line-card** role, it is not eligible to function in the master or backup role.
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 master switch. Do not power on the other switches at this time.
5. Run the EZSetup program on the master 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 master switch apply to the entire Virtual Chassis configuration, including all the members listed in the preprovisioned configuration file.

6. (Optional) Configure the master 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/
```
- Specify the preprovisioned configuration mode:

```
[edit virtual-chassis]
user@switch# set preprovisioned
```
  - 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
```
  - (Optional. Recommended for a two-member Virtual Chassis) Disable the split and merge feature:

```
[edit virtual-chassis]
user@switch# set no-split-detection
```
  - Power on the member switches.



**NOTE:** You cannot modify the mastership priority when you are using a preprovisioned configuration. The mastership priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file. The two Routing Engines are assigned the same mastership priority value. However, the member that was powered on first has higher prioritization according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#).

---

## 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:

- 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.

- Power on only the switch that you plan to use as the master switch (SWA-0). Do not power on the other switches at this time.
- 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 mastership priority for the master, 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@switch# 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 mastership priority value for each member switch is 128. The master role is selected by default. You can change the role that is performed by the members by modifying the mastership priority. See [“Configuring Mastership of a Virtual Chassis \(CLI Procedure\)” on page 185](#). We recommend that you specify the same mastership priority value for the desired master and backup members. In this example, the highest possible mastership priority has been assigned to two members. However, the member that was powered on first has higher prioritization according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#). The other members use the default mastership priority in this example, which configures them to function in the role of linecard.



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

#### Related Documentation

- [Example: Configuring an EX3300 Virtual Chassis with a Master and Backup on page 54](#)
- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 185](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)

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

---

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 165.

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 Switch as a Virtual Chassis Port \(CLI Procedure\)”](#) on page 188 or [“Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\)”](#) on page 193.



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

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 master sequentially assigns a member ID to other member switches. The role is determined by the mastership priority value and other factors in the master election algorithm.

This topic includes:

- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis with a Preprovisioned Configuration File](#) on page 156
- [Configuring an EX4200, an EX4500, or EX4550 Virtual Chassis with a Nonprovisioned Configuration File](#) on page 158

### 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 master or backup. If you configure the member with a **line-card** role, it is not eligible to become a master 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 master switch. Do not power on the other switches at this time.
5. Run the EZSetup program on the master 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 master switch apply to the entire Virtual Chassis configuration, including all the members listed in the preprovisioned configuration file.

6. (Optional) Configure the master 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 master 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 mastership priority when you are using a preprovisioned configuration. The mastership priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file. The two Routing Engines are assigned the same mastership priority value. However, the member that was powered on first has higher prioritization according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#).

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 master 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 mastership priority for the master, backup, and other members, if desired:

```
[edit virtual-chassis]
user@switch# set member 0 mastership-priority 255
user@switch# 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@switch# 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 mode 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 mastership priority value for each member switch is 128. The master role is selected by default. You can change the role that is performed by the members by modifying the mastership priority. See [“Configuring Mastership of a Virtual Chassis \(CLI Procedure\)” on page 185](#). We recommend that you specify the same mastership priority value for the desired master and backup members. In this example, the highest possible mastership priority has been assigned to two members. However, the member that was powered on first has higher prioritization according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#). The other members use the default mastership priority in this example, which configures them to function in the role of linecard.



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

Related Documentation

- [Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis on page 116](#)
- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 185](#)

- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)

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

- [Configuring an EX2200, EX2200-C, EX3300, EX4200, EX4300, EX4500, EX4550, or EX4600 Virtual Chassis \(J-Web Procedure\) on page 160](#)
- [Enabling Virtual Chassis Mode on an EX8200 Switch \(J-Web Procedure\) on page 162](#)
- [Configuring an EX8200 Virtual Chassis \(J-Web Procedure\) on page 162](#)

[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.

J-Web is supported on EX4600 switches only in J-Web Application package Release 14.1X53-A2.

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 Switch as a Virtual Chassis Port \(CLI Procedure\)” on page 188](#).



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 master, 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 mastership 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 25 on page 161](#).
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 25: Virtual Chassis Configuration Fields

Field	Function	Your Action
Member Details		

Table 25: Virtual Chassis Configuration Fields (*continued*)

Field	Function	Your Action
Member ID	Specifies the identifier for the member switch.	Select an identifier (from 0 through 9) from the list. NOTE: For EX2200 Virtual Chassis, you can select the member ID (from 0 through 3) from the list.
Priority	Specifies the mastership priority to be assigned to the member.	Select a number from 1 through 255 , (255 being the highest priority and 128 , 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)

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 on page 163](#)
- [Configure Virtual Chassis Members on page 163](#)
- [Configure Virtual Chassis Ports on page 163](#)

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.

Related Documentation

- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) on page 193](#)
- [Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets on page 91](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)
- [Virtual Chassis Cabling Configuration Examples for EX4200 Switches](#)
- [EX Series Virtual Chassis Overview on page 3](#)
- [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 156.

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 master, 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 master sequentially assigns a member ID to other member switches and the other member switches roles are determined by the mastership priority value and other factors in the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected”](#) on page 20.

Ensure that all switches are running the same version of Junos OS. See *Installing Software on an EX Series Switch with a 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 master and backup roles. See [“Understanding EX Series Virtual Chassis Components”](#) on page 9.
3. Power and log in to the switch that you want to function in the master 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 master switch.
11. Run the EZSetup program on the master switch, specifying the identification parameters. See *Connecting and Configuring an EX Series Switch (CLI Procedure)*.



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

12. (Optional) On the master 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 master switch, specify the preprovisioned configuration mode:

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

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

14. On the master 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 Switch as a Virtual Chassis Port \(CLI Procedure\)](#)” on page 188 or “[Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\)](#)” on page 193).



NOTE: You cannot modify the mastership priority when you are using a preprovisioned configuration. The mastership priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file.

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 Mastership of a Virtual Chassis \(CLI Procedure\) on page 185](#)
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\) on page 171](#)
- [Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) on page 176](#)
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 178](#)

- [Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 180](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 188](#)
- [Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) on page 193](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)

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. See *Downloading Software Packages from Juniper Networks*.



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 master 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 abort 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 \(CLI Procedure\) on page 165](#)
- [Understanding EX Series Virtual Chassis Components on page 9](#)

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 master Virtual Chassis configuration file are applied to the new switch after it has been interconnected to an existing member switch.



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

- (Optional) Configured Ethernet interfaces on different member switches into the same LAG. See [“Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch” on page 120](#)

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 on page 67](#)

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

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 \(CLI Procedure\)” on page 176](#).

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 on page 171](#)
- [Adding a New Switch from a Different Wiring Closet to an Existing Virtual Chassis on page 173](#)
- [Adding a New Switch to an Existing Preprovisioned Virtual Chassis Using Autoprovisioning on page 175](#)

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 master 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” on page 120](#)

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 master 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 master 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 mastership 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

This procedure can be used to add an EX4200 switch to an existing EX4200 Virtual Chassis using autoprovisioning.

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 master. For more information, see [“Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File”](#) on page 104.
- On the master, 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. 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 master 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 configuring an uplink module port as a VCP on the existing member switch.
- 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 Switch as a Virtual Chassis Port \(CLI Procedure\)”](#) on page 188.

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 on page 71](#)
- [Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration on page 85](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets on page 91](#)
- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104](#)
- [Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches on page 144](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)
- [Replacing a Member Switch of a Virtual Chassis Configuration \(CLI Procedure\) on page 181](#)

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 Switch as a Virtual Chassis Port \(CLI Procedure\)](#)” on page 188.

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 Mastership of a Virtual Chassis \(CLI Procedure\) on page 185](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 188](#)
- [Setting an Uplink Port as a Virtual Chassis Port on an EX4500 or EX4550 Switch \(CLI Procedure\) on page 193](#)

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 master 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 master 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 \(CLI Procedure\) on page 180](#)
- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 185](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 188](#)

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 193](#).

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 mastership priority of the EX4500 switch. The switches with the highest mastership priorities assume the master role and backup roles.

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

```
user@switch# set member member-id mastership-priority mastership-priority
```

14. (Optional) From the master switch, set the mastership 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 on page 81](#)
- [Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) on page 176](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)

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



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

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

If you want to replace a member switch of a *mixed* Virtual Chassis that contains EX4200, EX4500, or EX4550 switches, see [“Removing an EX4200, EX4500, or EX4550 Switch From a Mixed Virtual Chassis \(CLI Procedure\)”](#) on page 184.

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

- [Remove, Repair, and Reinstall the Same Switch on page 182](#)
- [Remove a Member Switch, Replace It with a Different Switch, and Reapply the Old Configuration on page 182](#)
- [Remove a Member Switch and Make Its Member ID Available for Reassignment to a Different Switch on page 184](#)

Remove, Repair, and Reinstall the Same Switch

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

To remove, repair, and reinstall the member switch:

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

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

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



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

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

1. Power off and disconnect the member switch to be replaced.
2. If the replacement member switch has been previously configured, revert that switch's configuration to the factory defaults. See *Reverting to the Default Factory Configuration for the EX Series Switch* for information about reverting to the factory default configuration on an EX Series switch or *Reverting to the Default Factory Configuration*

for information about reverting to the factory default configuration on a QFX Series switch.

3. (Required when automatic software update is not enabled on the Virtual Chassis and the new member switch is running a version of Junos OS that is different than the version of Junos OS running on the Virtual Chassis) Ensure that the correct version of Junos OS is or will be installed on the new member switch by performing one of the following tasks:

- Enable automatic software update on the Virtual Chassis. See [“Configuring Automatic Software Update on Virtual Chassis Member Switches \(CLI Procedure\)” on page 199](#).
- Install the version of Junos OS running on the Virtual Chassis onto the new member switch before interconnecting it into the Virtual Chassis. See *Installing Software on an EX Series Switch with a Single Routing Engine (CLI Procedure)*.

4. If you are interconnecting a switch using a dedicated VCP, connect one VCP on the replacement member switch to a VCP of another Virtual Chassis member switch.

If you are interconnecting a switch using an optical port configured as a VCP, cable the optical ports together then configure the port on the Virtual Chassis as a VCP:

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

5. Power on the new member switch.
6. Confirm that the new member switch is now included in the Virtual Chassis configuration by checking the front-panel LCD or the for the member ID. It should display a member ID in the range from 0 through 9.

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

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



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

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

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

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

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



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

Related Documentation

- [Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis \(CLI Procedure\)](#)
- [Adding a New Switch to an Existing QFX Series Virtual Chassis \(CLI Procedure\)](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\) on page 171](#)
- [Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) on page 176](#)
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 178](#)
- [Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 180](#)

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 [“Replacing a Member Switch of a Virtual Chassis Configuration \(CLI Procedure\)” on page 181](#).

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 [“Replacing a Member Switch of a Virtual Chassis Configuration \(CLI Procedure\)” on page 181](#). If the Virtual Chassis was nonprovisioned,

change the mastership-priority values of each member switch as needed to reconfigure the Virtual Chassis roles. See [“Configuring Mastership of a Virtual Chassis \(CLI Procedure\)”](#) on page 185.

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 Mastership of a Virtual Chassis (CLI Procedure)



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

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



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

This topic describes:

- [Configuring Mastership Using a Preprovisioned Configuration File](#) on page 186
- [Configuring Mastership Using a Configuration File That Is Not Preprovisioned](#) on page 186

Configuring Mastership Using a Preprovisioned Configuration File

To configure mastership using a preprovisioned configuration:

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

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

4. Specify the serial numbers of the member switches that you want to function as master and backup, specifying their role as **routing-engine**:

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



NOTE: You cannot directly modify the mastership priority value when you are using a preprovisioned configuration. The mastership priority values are generated automatically and controlled by the role that is assigned to the member switch in the configuration file. The two members assigned the **routing-engine** role are assigned the same mastership priority value (128). However, the member that was powered on first has higher priority for the master role election according to the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#). Only two members can be configured with the **routing-engine** role.

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

Configuring Mastership Using a Configuration File That Is Not Preprovisioned

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

1. Power on only the switch that you want to function in the master role.
2. Configure the highest possible mastership priority value (**255**) for the member that you want to function in the master role:

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

3. Configure the same mastership priority value (continue to edit the Virtual Chassis configuration on the master) for the member that you want to be in the backup role:

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



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

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

Related Documentation

- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring a QFX Series Virtual Chassis \(CLI Procedure\)](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\) on page 171](#)
- [Adding an EX4200 Switch to a Preprovisioned EX4500 Virtual Chassis or a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis \(CLI Procedure\) on page 176](#)
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 178](#)
- [Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 180](#)

Setting a Port on an EX2200 Switch as a Virtual Chassis Port (CLI Procedure)

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. 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.

To set a port on an EX2200 or EX2200-C switch as a VCP:

```
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
```

You can configure an interface of a different member switch as a VCP when both switches are part of the same operational Virtual Chassis. You can perform this task by entering the same command with the *member* option:

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

For instance, if you wanted to set built-in port 20 on member switch 3 as a VCP:

```
user@switch> request virtual-chassis vc-port set pic-slot 0 port 20 member 3
```



NOTE: 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 EX Series Virtual Chassis Port Link Aggregation”](#) on page 26.

**Related
Documentation**

- [Configuring an EX2200 Virtual Chassis \(CLI Procedure\) on page 148](#)

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

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

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

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



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” on page 335](#) for more information.

You must manually configure VCPs to connect EX2200 switches together to form an EX2200 Virtual Chassis. See [“Setting a Port on an EX2200 Switch as a Virtual Chassis Port \(CLI Procedure\)” on page 187](#).

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

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

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



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



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

Before you set an uplink port as a VCP:

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



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

3. Run EZSetup on the switch that you are configuring to be the master. Follow the prompts to specify the hostname and other identification, time zone, and network properties. See *Connecting and Configuring an EX Series Switch (CLI Procedure)* for details. The properties that you specify for the master apply to the entire Virtual Chassis, including all the member switches that you later interconnect with the master.
4. If you want to configure and manage the Virtual Chassis remotely, specify the VME global management interface. You can configure the VME global management interface when you are setting up the master or you can do it after completing the other configuration steps for the Virtual Chassis. See [“Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis \(CLI Procedure\)” on page 195](#).
5. Configure mastership of the Virtual Chassis by using either the nonprovisioned or preprovisioned configuration. See [“Configuring Mastership of a Virtual Chassis \(CLI Procedure\)” on page 185](#) for details.



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

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

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



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

This topic describes:

1. [Setting an Uplink VCP Between the Member Switches on page 190](#)
2. [Setting an Uplink VCP on a Standalone Switch on page 191](#)

Setting an Uplink VCP Between the Member Switches

You can set an uplink port as a VCP.



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

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

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

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

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

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

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

Setting an Uplink VCP on a Standalone Switch

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

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

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

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



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

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

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



.....

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

.....

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

**Related
Documentation**

- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) on page 152](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)
- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets on page 91](#)
- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)

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-replaceable 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” on page 335](#) 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 on page 98](#)

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 \(CLI Procedure\) on page 156](#)
- [Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port \(CLI Procedure\) on page 188](#)
- [Understanding Interface Naming Conventions on EX Series Switches](#)

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 master 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 master.
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 Master and Backup on page 54](#)
- [Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58](#)
- [Understanding Global Management of a Virtual Chassis on page 22](#)

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

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

The MAC persistence timer is used in situations in which the master switch is no longer a member of the Virtual Chassis because it has been physically disconnected or removed. If the old master switch does not rejoin the Virtual Chassis before the timer elapses, the new master switch starts using its own MAC address as the system's MAC base address. For information regarding how the system MAC base address is used to assign MAC addresses to ports in a Virtual Chassis, see [“Understanding MAC Address Assignment on a Virtual Chassis” on page 43](#).

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

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

To configure or modify the MAC persistence timer:

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

To disable the MAC persistence timer:

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

Related Documentation

- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) on page 152](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a QFX Series Virtual Chassis \(CLI Procedure\)](#)
- [Understanding EX Series Virtual Chassis Components on page 9](#)

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 reenabling 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

- [Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails on page 131](#)
- [Disabling Fast Failover in an EX Series Virtual Chassis on page 198](#)
- [Understanding Fast Failover in an EX Series Virtual Chassis on page 30](#)

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

- [Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails on page 131](#)
- [Configuring Fast Failover in an EX Series Virtual Chassis on page 197](#)
- [Understanding Fast Failover in an EX Series Virtual Chassis on page 30](#)

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

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

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

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

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

```
[edit]
```

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

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

```
[edit]
```

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

Related Documentation

- [Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge on page 134](#)
- [Understanding Split and Merge in a Virtual Chassis on page 37](#)

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

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



NOTE: The version of Junos OS running on the Virtual Chassis must be compatible with the software running on the prospective member switch for an automatic software update to occur. For information on Junos OS compatibility and other automatic software update restrictions, see [“Understanding Automatic Software Update on Virtual Chassis Member Switches” on page 40](#).

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

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

[edit]

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

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

[edit]

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

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

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

/pathname/package-name

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

ftp://hostname/pathname/package-name

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

http://hostname/pathname/package-name

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

**Related
Documentation**

- [Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches on page 144](#)
- [Understanding Automatic Software Update on Virtual Chassis Member Switches on page 40](#)

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

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

To configure the Virtual Chassis ID:

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

Related Documentation

- [Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge on page 134](#)
- [Understanding Split and Merge in a Virtual Chassis on page 37](#)

Configuring Graceful Routing Engine Switchover in a Virtual Chassis (CLI Procedure)

In a Virtual Chassis, one member switch is assigned the master role and has the master Routing Engine. Another member switch is assigned the backup role and has the backup Routing Engine. Graceful Routing Engine switchover (GRES) enables the master and backup Routing Engines in a Virtual Chassis configuration to switch from the master to backup without interruption to packet forwarding. When you configure graceful Routing Engine switchover, the backup Routing Engine automatically synchronizes with the master 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 mastership 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 Master and Backup in a Single Wiring Closet on page 58](#)
 - [High Availability Features for EX Series Switches Overview](#)
 - [Understanding EX Series Virtual Chassis Configuration on page 28](#)
 - [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 on page 250](#)
- *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 (CLI Procedure)*

CHAPTER 4

Configuration Statements

- [\[edit virtual-chassis\] Configuration Statement Hierarchy on page 205](#)
- [auto-sw-update on page 208](#)
- [fast-failover on page 210](#)
- [graceful-restart \(Enabling Globally\) on page 211](#)
- [graceful-switchover on page 212](#)
- [id on page 213](#)
- [lag-hash on page 214](#)
- [location \(Virtual Chassis\) on page 215](#)
- [mac-persistence-timer on page 216](#)
- [mastership-priority on page 217](#)
- [member on page 219](#)
- [no-management-vlan on page 220](#)
- [no-split-detection on page 221](#)
- [package-name on page 222](#)
- [preprovisioned on page 223](#)
- [redundancy \(Graceful Switchover\) on page 224](#)
- [role on page 225](#)
- [serial-number on page 228](#)
- [traceoptions \(Virtual Chassis\) on page 229](#)
- [vc-port on page 232](#)
- [vcp-snmp-statistics on page 233](#)
- [virtual-chassis on page 234](#)

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

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

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

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

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

This topic lists:

- [Supported Statements in the \[edit virtual-chassis\] Hierarchy Level on page 206](#)
- [Unsupported Statements in the \[edit virtual-chassis\] Hierarchy Level on page 207](#)

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

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

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

}

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

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

Related Documentation

- *Preprovisioning a Virtual Chassis Fabric*
- *Autoprovisioning a Virtual Chassis Fabric*
- *Adding a Device to a Virtual Chassis Fabric*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
- *Configuring an EX4300 Virtual Chassis (CLI Procedure)*
- *Configuring an EX2200 Virtual Chassis (CLI Procedure) on page 148*
- *Configuring an EX3300 Virtual Chassis (CLI Procedure) on page 152*
- *Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) on page 156*
- *Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure) on page 165*
- *Configuring an EX8200 Virtual Chassis (CLI Procedure)*

auto-sw-update

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

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

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

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

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

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

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

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

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

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

Related Documentation	<ul style="list-style-type: none"> • Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches on page 144 • Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure) on page 199 • Understanding Software Upgrades in a Virtual Chassis Fabric
------------------------------	---

fast-failover

Syntax	<code>fast-failover (ge vcp disable xe);</code>
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 9.3 for EX Series switches.
Description	<p>Enable the fast failover feature on an uplink Virtual Chassis port (VCP).</p> <p>Fast failover is not supported on an EX2200 Virtual Chassis or EX8200 Virtual Chassis.</p>
Default	<p>Fast failover is enabled, by default, on dedicated VCPs.</p> <p>Fast failover is not enabled on uplink VCPs, including the uplink ports that are, by default, configured as VCPs on an EX3300 switch.</p>
Options	<ul style="list-style-type: none">• 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	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails on page 131• Configuring Fast Failover in an EX Series Virtual Chassis on page 197• Disabling Fast Failover in an EX Series Virtual Chassis on page 198

graceful-restart (Enabling Globally)

Syntax	<pre> graceful-restart { disable; helper-disable; maximum-helper-recovery-time <i>seconds</i>; maximum-helper-restart-time <i>seconds</i>; notify-duration <i>seconds</i>; recovery-time <i>seconds</i>; restart-duration <i>seconds</i>; stale-routes-time <i>seconds</i>; }</pre>
Hierarchy Level	<pre> [edit logical-systems <i>logical-system-name</i> routing-options], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> routing-options], [edit routing-options], [edit routing-instances <i>routing-instance-name</i> routing-options]</pre>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced for QFX switches in Junos OS Release 13.2X51-D15</p> <p>Statement introduced for QFX Virtual Chassis and Virtual Chassis Fabric in Junos OS Release 14.1X53-D30</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	<p>Configure graceful restart globally to enable the feature. You cannot enable graceful restart for specific protocols unless graceful restart is also enabled globally. You can, optionally, modify the global settings at the individual protocol level.</p>



NOTE:

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

Default	Graceful restart is disabled by default.
Options	The remaining statements are explained separately.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

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


graceful-switchover

- Syntax** graceful-switchover;
- Hierarchy Level** [edit chassis [redundancy](#)]
- Release Information** Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
- Description** For switches with more than one Routing Engine, including those in a Virtual Chassis or a Virtual Chassis Fabric, configure the master 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.
- Related Documentation**
- *Example: Configuring Nonstop Active Routing on Switches*
 - *Configuring Graceful Routing Engine Switchover*
 - [Configuring Graceful Routing Engine Switchover in a Virtual Chassis \(CLI Procedure\) on page 201](#)
 - *Configuring Nonstop Active Routing on Switches*
 - *Installing Software on an EX Series Switch with Redundant Routing Engines (CLI Procedure)*

id

Syntax	<code>id id;</code>
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 9.3 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Configure the alphanumeric string that identifies a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.
Options	<i>id</i> —Virtual Chassis ID (VCID), which uses the ISO family address format—for example, 9622.6ac8.5345 .
Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge on page 134 • Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure) on page 201 • Configuring a QFX Series Virtual Chassis (CLI Procedure) • 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

lag-hash

Syntax	lag-hash (packet-based source-port-based);
Hierarchy Level	[edit virtual-chassis vc-port]
Release Information	Statement introduced in Junos OS Release 12.1 for EX Series switches.
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.
	<div> 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.</div>
Default	source-port-based
Options	<p>packet-based—Hashes all incoming LAG network traffic on the dedicated trunk port based on the packet.</p> <p>source-port-based—Hashes all incoming LAG network traffic on the dedicated trunk port based on the source.</p>
Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• <i>Understanding Link Aggregation in an EX8200 Virtual Chassis</i>

location (Virtual Chassis)

Syntax	<code>location location;</code>
Hierarchy Level	[edit virtual-chassis member member-id]
Release Information	Statement introduced in Junos OS Release 11.1 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	<p>Set a description of the location of the Virtual Chassis or VCF member switch or external Routing Engine.</p> <p>The Location field is visible to users who enter the show virtual-chassis status detail command.</p> <p>Setting this description has no effect on the operation of the member device.</p>
Options	location —Location of the current member switch or external Routing Engine. The location can be any single word.
Required Privilege Level	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Autoprovisioning a Virtual Chassis Fabric</i> • <i>Preprovisioning a Virtual Chassis Fabric</i> • <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i> • Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104 • Example: Configuring a Preprovisioned Mixed EX4200 and EX4500 Virtual Chassis on page 116 • <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i> • Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) on page 156 • Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure) on page 165 • <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i>

mac-persistence-timer

Syntax	<code>mac-persistence-timer [<i>minutes</i> disable];</code>
Hierarchy Level	[edit virtual-chassis]
Release Information	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option disable introduced in Junos OS Release 12.2 for EX Series switches.</p> <p>The maximum timer limit changed from no maximum timer limit to 60 minutes in Junos OS Release 12.2 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>Specify how long the Virtual Chassis or VCF continues to use the MAC address of the switch that was originally configured in the master role as the system MAC base address after the original master switch is removed from the Virtual Chassis or VCF. The system MAC base address does not change in the event of a switchover provided the switch originally configured in the master role remains a member of the Virtual Chassis or VCF.</p> <p>The maximum timer limit is 60 minutes starting in Junos OS Release 12.2. There are no minimum or maximum timer limits in prior Junos OS releases.</p>
Default	The MAC persistence timer is set to 10 minutes by default.
Options	<p>minutes—Time in minutes that the member switch in the backup role continues to use the system MAC base address of the old master before using its own system MAC base address after the switch in the master role is physically disconnected or removed from the Virtual Chassis or VCF.</p> <p>disable—Disable the MAC persistence timer. The system MAC base address never changes when the MAC persistence timer is disabled, even when the switch in the master role is physically disconnected or removed from the Virtual Chassis or VCF.</p>
Required Privilege Level	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Configuring the Timer for the Backup Member to Start Using Its Own MAC Address, as Master of a Virtual Chassis (CLI Procedure) on page 196• <i>Autoprovisioning a Virtual Chassis Fabric</i>• <i>Preprovisioning a Virtual Chassis Fabric</i>

mastership-priority

Syntax	<code>mastership-priority <i>number</i>;</code>
Hierarchy Level	[edit virtual-chassis member <i>member-id</i>]
Release Information	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Mastership priority option 0 introduced in Junos OS Release 11.1 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>The mastership priority value is the most important factor in determining the role of the member switch within a nonprovisioned Virtual Chassis or VCF configuration. Other factors (see “Understanding How the Master in a Virtual Chassis Is Elected” on page 20) also affect the election of the master.</p> <p>The mastership priority value takes the highest precedence in the master election algorithm. The member switch with highest mastership priority assumes the master Routing Engine role of the Virtual Chassis or VCF. Toggling back and forth between master and backup status in failover conditions is undesirable, so we recommend that you assign the same mastership priority value to both the master and the backup. Secondary factors in the master election algorithm determine which of these two members (that is, the two members that are assigned the highest mastership priority value) functions as the master of the Virtual Chassis or VCF.</p> <p>This statement is not used for the EX8200 Virtual Chassis, which determines mastership by external Routing Engine uptime. See <i>Understanding Virtual Chassis Roles in an EX8200 Virtual Chassis</i>.</p> <p>A switch with a mastership priority of 0 never takes the master or backup role.</p>
Default	128
Options	<p><i>number</i>—Mastership priority value.</p> <p>Range: 0 through 255</p>
Required Privilege Level	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Autoprovisioning a Virtual Chassis Fabric • Preprovisioning a Virtual Chassis Fabric • Configuring a QFX Series Virtual Chassis (CLI Procedure) • Configuring an EX4300 Virtual Chassis (CLI Procedure) • Example: Configuring an EX3300 Virtual Chassis with a Master and Backup on page 54 • Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58

- [Example: Configuring an EX4200 Virtual Chassis Interconnected Across Multiple Wiring Closets on page 91](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)

member

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

- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104](#)
- [Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines](#)
- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) on page 152](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring an EX9200 Virtual Chassis](#)
- [Configuring a QFX Series Virtual Chassis \(CLI Procedure\)](#)

no-management-vlan

Syntax	no-management-vlan;
Hierarchy Level	[edit virtual-chassis member member-id]
Release Information	Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	<p>Remove the specified member's out-of-band management port from the virtual management Ethernet (VME) global management VLAN of the Virtual Chassis or VCF configuration.</p> <p>For a member that is functioning in a linecard role, you can use this configuration to reserve the member's management Ethernet port for local troubleshooting:</p> <pre>virtual-chassis { member 2 { no-management-vlan; } }</pre> <p>You cannot configure the IP address for a local management Ethernet port using the CLI or the J-Web interface. To do this, you need to use the shell ifconfig command.</p>
Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Setting Up a Multimember EX4200 Virtual Chassis Access Switch with a Default Configuration on page 85• Configuring the Virtual Management Ethernet Interface for Global Management of an EX Series Virtual Chassis (CLI Procedure) on page 195• Understanding Global Management of a Virtual Chassis on page 22• Understanding Virtual Chassis Fabric Configuration

no-split-detection

Syntax	no-split-detection;
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 9.3 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	<p>Disable the split and merge feature in a Virtual Chassis or VCF configuration.</p> <p>We recommend using this statement to disable the split and merge feature when configuring a two-member Virtual Chassis. Enabling this statement on a two-member Virtual Chassis ensures that both switches remain in the correct Virtual Chassis roles in the event of a Virtual Chassis split.</p> <p>The split and merge feature is enabled by default when a Virtual Chassis is initially set up. If you set this option in a two-member Virtual Chassis and then expand the Virtual Chassis to add more members, we recommend deleting this configuration item to re-enable the split and merge feature again.</p>
Default	The split and merge feature is enabled.
Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Example: Assigning the Virtual Chassis ID to Determine Precedence During an EX4200 Virtual Chassis Merge on page 134 • Disabling Split and Merge in a Virtual Chassis (CLI Procedure) on page 199 • Assigning the Virtual Chassis ID to Determine Precedence During a Virtual Chassis Merge (CLI Procedure) on page 201 • Disabling Split Detection in a Virtual Chassis Configuration • Understanding Split and Merge in a Virtual Chassis on page 37

package-name

Syntax	<code>package-name <i>package-name</i>;</code>
Hierarchy Level	[edit virtual-chassis auto-sw-update]
Release Information	Statement introduced in Junos OS Release 10.0 for EX Series switches. Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Specify the software package name or location of the software package to be used by the automatic software update feature for Virtual Chassis or VCF.
Default	No package name is specified.
Options	<p><i>package-name</i>—Name of the software package or the URL to the software package to be used.</p> <ul style="list-style-type: none">• If the software package is located on a local directory on the switch, use the following format for <i>package-name</i>: <i>/pathname/package-name</i>• If the software package is to be downloaded and installed from a remote location, use one of the following formats: <i>ftp://hostname/pathname/package-name</i> <i>ftp://username:prompt@ftp.hostname.net/package-name</i> <i>http://hostname/pathname/package-name</i>
Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring Automatic Software Update on EX4200 Virtual Chassis Member Switches on page 144• Configuring Automatic Software Update on Virtual Chassis Member Switches (CLI Procedure) on page 199• Understanding Software Upgrades in a Virtual Chassis Fabric

preprovisioned

Syntax	preprovisioned;
Hierarchy Level	[edit virtual-chassis]
Release Information	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>Enable the preprovisioned configuration mode for a Virtual Chassis or Virtual Chassis Fabric (VCF) configuration.</p> <p>When the preprovisioned configuration mode is enabled, you cannot use the CLI or the J-Web interface to change the mastership priority or member ID of member switches.</p> <p>You must use this statement to configure an EX8200 Virtual Chassis. Nonprovisioned configuration of an EX8200 Virtual Chassis is not supported.</p>
Required Privilege Level	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Preprovisioning a Virtual Chassis Fabric</i> • Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104 • <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i> • Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) on page 156 • <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i> • <i>Configuring an EX9200 Virtual Chassis</i> • <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i> • Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure) on page 181

redundancy (Graceful Switchover)

Syntax	<pre>redundancy { failover { on-disk-failure; on-loss-of-keepalives; } graceful-switchover; }</pre>
Hierarchy Level	[edit chassis]
Release Information	<p>Statement introduced in Junos OS Release 9.2 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	<p>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.</p> <p>The remaining statements are explained separately.</p>
Default	Redundancy is enabled for the Routing Engines.
Required Privilege Level	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• graceful-switchover on page 212• Configuring Graceful Routing Engine Switchover in a Virtual Chassis (CLI Procedure) on page 201• <i>Configuring Graceful Routing Engine Switchover</i>• <i>Installing Software on an EX Series Switch with Redundant Routing Engines (CLI Procedure)</i>• <i>High Availability Features for EX Series Switches Overview</i>

role

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

Virtual Chassis Fabric

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

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

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

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

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

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

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

EX8200 Virtual Chassis

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

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

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

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

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

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

(All Virtual Chassis composed of EX Series switches, except EX8200 switches, or QFX Series devices) Specify two and only two members as **routing-engine**. The software determines which of the two members assigned the **routing-engine** role functions as master, based on the master election algorithm. See [“Understanding How the Master in a Virtual Chassis Is Elected” on page 20](#). In these Virtual Chassis, the **routing-engine** role is associated with a switch.

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

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

Related Documentation

- *Autoprovisioning a Virtual Chassis Fabric*
- *Preprovisioning a Virtual Chassis Fabric*
- [Example: Configuring an EX4200 Virtual Chassis Using a Preprovisioned Configuration File on page 104](#)
- *Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines*
- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) on page 152](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- *Configuring an EX8200 Virtual Chassis (CLI Procedure)*
- *Configuring an EX9200 Virtual Chassis*
- *Configuring a QFX Series Virtual Chassis (CLI Procedure)*
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)
- [Adding a New EX4200 Switch to an Existing EX4200 Virtual Chassis \(CLI Procedure\) on page 171](#)
- [Replacing a Member Switch of a Virtual Chassis Configuration \(CLI Procedure\) on page 181](#)

serial-number

Syntax	<code>serial-number serial-number;</code>
Hierarchy Level	[edit virtual-chassis preprovisioned member member-id]
Release Information	<p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>In a preprovisioned Virtual Chassis or Virtual Chassis Fabric (VCF), specify the serial number of each member switch to be included in the configuration. If you do not include the serial number within the configuration, the switch cannot be recognized as a member of a preprovisioned configuration.</p> <p>In an EX8200 Virtual Chassis configuration, specify the serial number of each XRE200 External Routing Engine and each EX8200 member switch to be included in the Virtual Chassis configuration. If you do not include the serial number within the Virtual Chassis configuration, the external Routing Engine or switch cannot be recognized as a member of the configuration.</p>
Options	serial-number —Permanent serial number for the external Routing Engine or for the member switch.
Required Privilege Level	<p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Autoprovisioning a Virtual Chassis Fabric• Preprovisioning a Virtual Chassis Fabric• Configuring an EX2200 Virtual Chassis (CLI Procedure) on page 148• Configuring an EX3300 Virtual Chassis (CLI Procedure) on page 152• Configuring an EX4300 Virtual Chassis (CLI Procedure)• Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) on page 156• Configuring an EX8200 Virtual Chassis (CLI Procedure)• Configuring an EX9200 Virtual Chassis• Configuring a QFX Series Virtual Chassis (CLI Procedure)• Configuring a Virtual Chassis on an EX Series Switch (J-Web Procedure) on page 160

tracoptions (Virtual Chassis)

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

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

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

Description Define tracing operations for the Virtual Chassis or VCF.

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



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

Default Tracing operations are disabled.

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



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

disable—(Optional) Disable a flag.

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

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

Range: 2 through 1000

Default: 3 files

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

- **all**—All tracing operations.



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

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

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

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

receive—(Optional) Trace received packets.

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

send—(Optional) Trace transmitted packets.

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

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

Range: 10 KB through 1 GB

Default: 128 KB

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

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

Related Documentation	<ul style="list-style-type: none">• Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247• Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member on page 244• Verifying That Virtual Chassis Ports Are Operational on page 245• Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis• Troubleshooting an EX Series Virtual Chassis on page 335• Troubleshooting Virtual Chassis Fabric
------------------------------	--

vc-port

Syntax `vc-port {
 lag-hash (packet-based | source-port-based);
 }`

Hierarchy Level [edit `virtual-chassis`]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description Enable hashing of link aggregation group (LAG) network traffic over a dedicated trunk port within a Virtual Chassis.

You select how to direct all LAG traffic through the dedicated trunk port by using the `lag-hash` statement.



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.

The remaining statement is explained separately.

Default source-port-based

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

Related Documentation • *Understanding Link Aggregation in an EX8200 Virtual Chassis*

vcp-snmp-statistics

Syntax	vcp-snmp-statistics;
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 14.1X53-D15 for EX Series switches. Statement introduced in Junos OS Release 14.1X53-D15 for the QFX Series.
Description	<p>Enable SNMP monitoring of the Virtual Chassis ports (VCPs) for all VCPs in the Virtual Chassis or Virtual Chassis Fabric (VCF).</p> <p>When this statement is enabled, SNMP gathers statistics on the Junos VCP MIBs. You can retrieve the statistics gathered by SNMP for these MIBs by using the show snmp mib command with the walk and ascii options and specifying jnxVirtualChassisPortInPkts.</p>
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.
Related Documentation	<ul style="list-style-type: none"> • <i>Understanding EX4300 Virtual Chassis</i> • <i>Understanding QFX Series Virtual Chassis</i> • Understanding EX Series Virtual Chassis Components on page 9 • <i>Understanding QFX Series Virtual Chassis Components</i>

virtual-chassis

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

Hierarchy Level [edit]

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

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

The remaining statements are explained separately.

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

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

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

Required Privilege Level	system—To view this statement in the configuration. system-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• <i>Autoprovisioning a Virtual Chassis Fabric</i>• <i>Preprovisioning a Virtual Chassis Fabric</i>• <i>Adding a Device to a Virtual Chassis Fabric</i>• <i>Configuring a QFX Series Virtual Chassis (CLI Procedure)</i>• Example: Configuring an EX3300 Virtual Chassis with a Master and Backup on page 54• Example: Configuring an EX4200 Virtual Chassis with a Master and Backup in a Single Wiring Closet on page 58• <i>Example: Setting Up a Full Mesh EX8200 Virtual Chassis with Two EX8200 Switches and Redundant XRE200 External Routing Engines</i>• Configuring an EX3300 Virtual Chassis (CLI Procedure) on page 152• Configuring an EX4200, EX4500, or EX4550 Virtual Chassis (CLI Procedure) on page 156• <i>Configuring an EX8200 Virtual Chassis (CLI Procedure)</i>• <i>Configuring an EX9200 Virtual Chassis</i>

PART 3

Administration

- [Routine Monitoring on page 239](#)
- [Operational Commands on page 253](#)

CHAPTER 5

Routine Monitoring

- [Command Forwarding Usage with an EX Series Virtual Chassis on page 239](#)
- [Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member on page 244](#)
- [Verifying That Virtual Chassis Ports Are Operational on page 245](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)
- [Verifying That Graceful Routing Engine Switchover Is Working in the Virtual Chassis on page 249](#)
- [Verifying the Setting for the PIC Mode on an EX4500 Switch in a Virtual Chassis on page 250](#)
- [Verifying the Setting for the Virtual Chassis Mode on EX4200, EX4500, and EX4550 Switches on page 250](#)
- [Verifying Connectivity Between Virtual Chassis Member Devices on page 251](#)

Command Forwarding Usage with an EX Series Virtual Chassis

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

You can always specify that these commands be applied to all member switches in the Virtual Chassis by using the **all-members** option, or to a specific member switch by using the **member-member-id** option. If neither option is specified, the default command forwarding behavior, which varies by command, is used. See the **Default** row in [Table 26 on page 240](#) 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 26 on page 240](#) 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 26: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
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"> • show version • show chassis firmware • show chassis hardware • show chassis environment • show interfaces extensive (for each configured interface) • show configuration (excluding any SECRET-DATA) • show system virtual-memory 	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members
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 /altroot and /altconfig 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 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

Table 26: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration (*continued*)

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
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
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.	master 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
show system boot-messages	Display initial messages generated by the system kernel upon startup. These messages are the contents of <code>/var/run/dmesg.boot</code> .	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	all-members

Table 26: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration (*continued*)

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
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
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.	master 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 26: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration (*continued*)

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
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 <code>/altroot</code> and <code>/altconfig</code> file systems. To back up software, use the <code>request system snapshot</code> command.	Displays information for all members of the Virtual Chassis configuration.	Displays information for the specified member switch.	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
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

Table 26: Commands That Can be Run on All or Specific Members of the Virtual Chassis Configuration (*continued*)

Commands Available for Command Forwarding	Purpose	all-members	member-member-id	Default
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 on page 247](#)
 - [Understanding EX Series Virtual Chassis Components on page 9](#)
 - [Junos OS System Basics and Services Command Reference](#)

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

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

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

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

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

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

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

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


```

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

```

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

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



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

Related Documentation

- [Configuring Mastership of a Virtual Chassis \(CLI Procedure\) on page 185](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
- [Configuring a QFX Series Virtual Chassis \(CLI Procedure\)](#)
- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)

Verifying That Virtual Chassis Ports Are Operational

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



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

Action Display the VCPs:

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

```
fpc0:
```

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

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

fpc1:

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

fpc2:

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

fpc3:

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

fpc4:

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

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



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

Related Documentation

- [Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247](#)
- [Configuring an EX3300 Virtual Chassis \(CLI Procedure\) on page 152](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)

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

Purpose



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

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

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

Action

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



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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

**Related
Documentation**

- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring an EX8200 Virtual Chassis \(CLI Procedure\)](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
- [Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member on page 244](#)

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

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

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

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

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

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

1. Connect to the backup Routing Engine:

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

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

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

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

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

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



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

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

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



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

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

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

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

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

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

Related Documentation

- [Configuring Graceful Routing Engine Switchover in a Virtual Chassis \(CLI Procedure\) on page 201](#)

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

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 \(CLI Procedure\) on page 165](#)
- [Resetting the EX4500 Switch to Run as a Standalone Switch \(CLI Procedure\) on page 202](#)

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

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 \(CLI Procedure\) on page 165](#)

Verifying Connectivity Between Virtual Chassis Member Devices

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/Timedout/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/Timedout/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 on page 42](#)

CHAPTER 6

Operational Commands

- clear virtual-chassis vc-port statistics
- request chassis pic-mode
- request session member
- request virtual-chassis device-reachability
- request virtual-chassis mode
- request virtual-chassis recycle
- request virtual-chassis renumber
- request virtual-chassis vc-port
- request virtual-chassis vc-port
- show chassis pic-mode
- show snmp mib
- show system uptime
- show virtual-chassis active-topology
- show virtual-chassis device-topology
- show virtual-chassis protocol adjacency
- show virtual-chassis protocol database
- show virtual-chassis protocol interface
- show virtual-chassis protocol route
- show virtual-chassis protocol statistics
- show virtual-chassis fast-failover
- show virtual-chassis login
- show virtual-chassis mode
- show virtual-chassis
- show virtual-chassis vc-path
- show virtual-chassis vc-port
- show virtual-chassis vc-port statistics

clear virtual-chassis vc-port statistics

Syntax	<code>clear virtual-chassis vc-port statistics</code> <code><all-members></code> <code><interface-name></code> <code><local></code> <code><member member-id></code>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. The options all-members and local were added in Junos OS Release 9.3 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric.
Description	Clear—reset to zero (0)—the traffic statistics counters on Virtual Chassis ports (VCPs).
Options	none —Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF. all-members —(Optional) Clear traffic statistics for VCPs of all members of a Virtual Chassis or VCF. interface-name —(Optional) Clear traffic statistics for the specified VCP. local —(Optional) Clear traffic statistics for VCPs from the switch or external Routing Engine on which this command is entered. member member-id —(Optional) Clear traffic statistics for VCPs from the specified member of a Virtual Chassis or VCF.
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none">• show virtual-chassis vc-port statistics on page 326• show virtual-chassis vc-port on page 322• Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247
List of Sample Output	clear virtual-chassis vc-port statistics (EX4200 Virtual Chassis) on page 254 clear virtual-chassis vc-port statistics (EX8200 Virtual Chassis) on page 255 clear virtual-chassis vc-port statistics member 3 on page 255

Sample Output

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

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

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

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

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

clear virtual-chassis vc-port statistics member 3

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

```
Cleared statistics on member 3
```

request chassis pic-mode

Syntax `request chassis pic-mode (intraconnect | virtual-chassis)`
 `<all-members>`
 `<local>`
 `<member member-id>`

Release Information Command introduced in Junos OS Release 11.1 for EX Series switches.

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 202.

Use the **show chassis pic-mode** 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.

virtual-chassis—Set the PIC mode to Virtual Chassis.

all-members—(Optional) Set the PIC mode for all members of the Virtual Chassis configuration. This is the default option.

local—(Optional) Set the PIC mode on the member switch where the command is issued.

member *member-id*—(Optional) Set the PIC mode on the specified member of the Virtual Chassis configuration.

Required Privilege Level

view

Related Documentation

- [Verifying the Setting for the PIC Mode on an EX4500 Switch in a Virtual Chassis on page 250](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
- [Adding an EX4500 Switch to a Preprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 178](#)
- [Adding an EX4500 Switch to a Nonprovisioned EX4200 Virtual Chassis \(CLI Procedure\) on page 180](#)

List of Sample Output [request chassis pic-mode virtual-chassis on page 257](#)
[request chassis pic-mode intraconnect on page 257](#)

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)
```

request session member

Syntax	<code>request session member <i>member-id</i></code>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Start a session with the specified member of a Virtual Chassis or a VCF.
Options	<i>member-id</i> —Member ID for the specific member of the Virtual Chassis or VCF.
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none">• member on page 219• Understanding EX Series Virtual Chassis Components on page 9• Understanding QFX Series Virtual Chassis Components

request virtual-chassis device-reachability

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Required Privilege Level system-control

Related Documentation • [Verifying Connectivity Between Virtual Chassis Member Devices on page 251](#)

Output Fields [Table 27 on page 260](#) 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 27: request virtual chassis device reachability Output Fields

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

Sample Output

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

```
user@switch> request virtual-chassis device-reachability test-name member0-to-member2
source-fpc 0 destination-fpc 2
```

```
Device Reachability Statistics:
```

```
Test Name           : member0-to-member2
```

```
Performing Test: 1
```

```
56 bytes from 0: session-id 0 seq-id 0
```

```
56 bytes from 0: session-id 0 seq-id 1
```

```
56 bytes from 0: session-id 0 seq-id 2
```

```
56 bytes from 0: session-id 0 seq-id 3
```

```
56 bytes from 0: session-id 0 seq-id 4
```

```
--- ping statistics ---
```

```
Session ID           : 0
```

```
Packets Count/Sent/Received/Sendfail : 5/5/5/0
```

```
Unknown/Timedout/Duplicate packets received : 0/0/0
```

```
Round-trip Min/Avg/Max : 181/807/3010 usec
```


request virtual-chassis mode

Syntax	<pre>request virtual-chassis mode fabric mixed <disable> <reboot> <all-members> <local> <member <i>member-id</i>></pre>
Release Information	<p>Command introduced in Junos OS Release 11.1 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for QFX Series devices.</p> <p>The fabric keyword introduced in Junos OS Release 13.2X51-D20 for EX Series switches and QFX Series devices in a Virtual Chassis Fabric (VCF).</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for VCF.</p>
Description	<p>Configure the mode for a device or multiple devices in a Virtual Chassis or a VCF.</p> <p>A device must be configured in fabric mode to participate as a member device in a VCF.</p> <p>A device must be configured in mixed mode when it is participating in a Virtual Chassis or a VCF with different types of devices.</p> <p>Do not enable the request virtual-chassis mode mixed command for a standalone device or for a member switch that is intended to remain in a non-mixed Virtual Chassis or VCF. Enabling this command reduces the maximum scaling numbers for some features on the switch, Virtual Chassis, or VCF.</p> <p>You do not need to configure mixed mode if the only devices in your Virtual Chassis are EX4500 and EX4550 switches.</p> <p>To avoid potential traffic disruptions and configuration issues for a mixed Virtual Chassis, we recommend configuring mixed mode on your device before cabling it into your Virtual Chassis. We recommend rebooting your device to complete this configuration procedure before interconnecting your device into the Virtual Chassis.</p> <p>To avoid potential traffic disruptions and configuration issues, we recommend configuring the fabric and, if applicable, the mixed mode settings on your device before cabling it into a VCF. We recommend rebooting your device to complete this configuration procedure before interconnecting your device into the VCF. You can change the fabric and mixed mode settings after the device has been added to a Virtual Chassis or VCF, however.</p> <p>If you set some of the devices in a mixed Virtual Chassis or VCF to mixed mode using this command but not others, 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 set the Virtual Chassis mode to mixed for all devices in the Virtual Chassis or VCF. You then need to reboot the devices that have been set into mixed mode to complete the procedure. The Virtual Chassis or VCF forms after the devices have rebooted.</p> <p>When you do not use this command to set any of the switches in a mixed EX4200 and EX4500 Virtual Chassis to mixed mode, a mixed EX4200 and EX4500 Virtual Chassis</p>

forms with one of the switches assuming the master role if the switches are running Junos OS Release 11.4 or later. All other switches in the mixed EX4200 and EX4500 Virtual Chassis are placed into the linecard role. If you experience this behavior, enter the **request virtual-chassis mode mixed all-members** command to set the Virtual Chassis mode to mixed for all switches in the Virtual Chassis. You will then need to reboot the switches to complete the procedure. The Virtual Chassis will form after all of the switches have rebooted.

The Virtual Chassis mode setting is maintained through reboots even though it is set in operational mode.

- Options**
- none**—Set the Virtual Chassis mode for all members of the Virtual Chassis or VCF.
 - all-members**—(Optional) Set the Virtual Chassis mode for all members of the Virtual Chassis or VCF.
 - disable**—Disable the Virtual Chassis fabric or mixed mode setting if it was previously enabled.
 - fabric**—Set the device into fabric mode so that the device can participate in a VCF.
 - local**—(Optional) Set the Virtual Chassis mode on the member device where the command is issued.
 - member *member-id***—(Optional) Set the Virtual Chassis mode to mixed on the specified member of the Virtual Chassis or VCF.
 - mixed**—Set the device into mixed mode so that the device can participate in a mixed Virtual Chassis or mixed VCF.



NOTE: You do not need to set mixed mode if the only devices in your Virtual Chassis are QFX3500 and QFX3600 devices.

You do not need to configure mixed mode if the only devices in your Virtual Chassis are EX4500 and EX4550 switches.

Required Privilege Level system-control

- Related Documentation**
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
 - [Verifying the Virtual Chassis Fabric Mode Settings](#)
 - [Verifying the Member ID, Role, and Neighbor Member Connections of a Virtual Chassis Member on page 244](#)

List of Sample Output [request virtual-chassis mode mixed on page 263](#)
[request virtual-chassis mode fabric mixed reboot on page 263](#)

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
```

request virtual-chassis recycle

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

Release Information Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description Make a previously used member ID available for reassignment.

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



NOTE: You must run this command from the Virtual Chassis member in the master 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

Related Documentation

- [request virtual-chassis renumber on page 265](#)
- [Replacing a Member Switch of a Virtual Chassis Configuration \(CLI Procedure\) on page 181](#)
- [Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis \(CLI Procedure\)](#)

List of Sample Output [request virtual-chassis recycle member-id 3 on page 264](#)
[request virtual-chassis recycle member-id 1 on page 264](#)

Sample Output

`request virtual-chassis recycle member-id 3`


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

Sample Output

`request virtual-chassis recycle member-id 1`

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

request virtual-chassis renumber

Syntax	<code>request virtual-chassis renumber member-id <i>old-member-id</i> new-member-id <i>new-member-id</i></code>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
Description	Renumber a member of a Virtual Chassis configuration.
<div>  NOTE: You must run this command from the Virtual Chassis member in the master role. </div>	
Options	<p><code>member-id <i>old-member-id</i></code>—Specify the ID of the member that you wish to renumber.</p> <p><code>new-member-id <i>new-member-id</i></code>—Specify an unassigned member ID.</p>
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none"> • request virtual-chassis recycle on page 264 • Replacing a Member Switch of a Virtual Chassis Configuration (CLI Procedure) on page 181 • Adding or Replacing a Member Switch or an External Routing Engine in an EX8200 Virtual Chassis (CLI Procedure)
List of Sample Output	request virtual-chassis renumber member-id 5 new-member-id 4 on page 265 request virtual-chassis renumber member-id 1 new-member-id 0 on page 265

Sample Output

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

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

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

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

request virtual-chassis vc-port

Syntax	request virtual-chassis vc-port set interface <i>vcp-interface-name</i> <member <i>member-id</i>> <disable>
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches.
Description	<p>Disable or enable a Virtual Chassis port (VCP) on a dedicated VCP.</p> <p>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.</p>
Options	<p>interface <i>vcp-interface-name</i>—Name of the interface to enable or disable.</p> <p>If you omit member <i>member-id</i> 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.</p> <p>member <i>member-id</i>—(Optional on EX4200, EX4500, and EX4550 switches; required on EX8200 switches) Enable or disable the specified VCP on the specified member of the Virtual Chassis configuration.</p> <p>This option must be specified when using this command to configure a VCP link between two EX8200 switches.</p> <p>disable—(Optional) Disable the specified VCP. If you omit this keyword, the command enables the dedicated VCP.</p>
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none">• request virtual-chassis vc-port on page 268• show virtual-chassis vc-port on page 322• show virtual-chassis vc-port statistics on page 326• clear virtual-chassis vc-port statistics on page 254• Understanding EX Series Virtual Chassis Components on page 9
List of Sample Output	<p>request virtual-chassis vc-port set interface vcp-0 disable (EX4200 Virtual Chassis) on page 267</p> <p>request virtual-chassis vc-port set fpc-slot 4 pic-slot 0 port 2 member 1 (EX8200 Virtual Chassis) on page 267</p> <p>request virtual-chassis vc-port set interface vcp-0 member 3 disable (EX8200 Virtual Chassis) on page 267</p>

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.

request virtual-chassis vc-port

Syntax	<code>request virtual-chassis vc-port set delete <fpc-slot <i>fpc-slot</i>> pic-slot <i>pic-slot</i> port <i>port-number</i> <member <i>member-id</i>></code>
Release Information	<p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Option fpc-slot introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>Enable or disable an optical port as a Virtual Chassis port (VCP).</p> <p>If you omit member <i>member-id</i>, this command defaults to enabling or disabling the uplink VCP or SFP network port configured as a VCP on the switch where the command is issued.</p> <p>On an EX3300 switch, uplink ports 2 and 3 are configured as VCPs by default. No other uplink ports on any other EX Series switches are configured as VCPs by default.</p> <p>You might experience a temporary traffic disruption immediately after creating or deleting a user-configured VCP in an EX8200 Virtual Chassis.</p>
Options	<p>pic-slot <i>pic-slot</i>—Number of the PIC slot for the port on the switch.</p> <p>port <i>port-number</i>—Number of the port that is to be enabled or disabled as a VCP.</p> <p>member <i>member-id</i>—(Optional) Enable or disable the specified VCP on the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none">• request virtual-chassis vc-port on page 266 (dedicated port)• show virtual-chassis vc-port on page 322• show virtual-chassis vc-port statistics on page 326• clear virtual-chassis vc-port statistics on page 254• <i>Virtual Chassis Port (VCP) Interface Names in an EX8200 Virtual Chassis</i>• Understanding EX Series Virtual Chassis Components on page 9• <i>Understanding QFX Series Virtual Chassis Components</i>
List of Sample Output	<p>request virtual-chassis vc-port set pic-slot 1 port 0 on page 269</p> <p>request virtual-chassis vc-port set pic-slot 1 port 1 member 3 on page 269</p> <p>request virtual-chassis vc-port delete pic-slot 1 port 1 member 3 on page 269</p>

Sample Output

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

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

To check the results of this command, use the **show virtual-chassis vc-port** command.

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

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


To check the results of this command, use the **show virtual-chassis vc-port** command.

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

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

To check the results of this command, use the **show virtual-chassis vc-port** command.

show chassis pic-mode

Syntax	show chassis pic-mode <all-members> <local> <member <i>member-id</i>>
Release Information	Command introduced in Junos OS Release 11.1 for EX Series switches.
Description	<p>Display the PIC mode on an EX4500 switch.</p> <p>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.</p> <div> NOTE: This PIC mode setting is not used on an EX4550 switch.</div> <p>The default PIC mode setting is virtual-chassis.</p> <p>Use the request chassis pic-mode command to configure the PIC mode setting.</p>
Options	<p>none—Display the PIC mode on the member switch where the command is issued.</p> <p>all-members—(Optional) Display the PIC mode for all members of the Virtual Chassis configuration.</p> <p>local—(Optional) Display the PIC mode on the member switch where the command is issued.</p> <p>member <i>member-id</i>—(Optional) Display the PIC mode of the specified member of the Virtual Chassis configuration.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• request chassis pic-mode on page 256• Resetting the EX4500 Switch to Run as a Standalone Switch (CLI Procedure) on page 202• Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure) on page 165
List of Sample Output	show chassis pic-mode on page 270
Sample Output	
show chassis pic-mode	
	user@switch> show chassis pic-mode

fpc0:

Pic Mode: Intraconnect

show snmp mib

Syntax	<code>show snmp mib (get get-next walk) (ascii decimal) <i>object-id</i></code>
Release Information	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>ascii and decimal options introduced in Junos OS Release 9.6.</p> <p>ascii and decimal options introduced in Junos OS Release 9.6 for EX Series switches.</p> <p>Command introduced in Junos OS Release 11.1 for the QFX Series.</p> <p>Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Display local Simple Network Management Protocol (SNMP) Management Information Base (MIB) object values.
Options	<p>get—Retrieve and display one or more SNMP object values.</p> <p>get-next—Retrieve and display the next SNMP object values.</p> <p>walk—Retrieve and display the SNMP object values that are associated with the requested object identifier (OID). When you use this option, the Junos OS displays the objects below the subtree that you specify.</p> <p>ascii—Display the SNMP object's string indices as an ASCII-key representation.</p> <p>decimal—Display the SNMP object values in the decimal (default) format. The decimal option is the default option for this command. Therefore, issuing the show snmp mib (get get-next walk) decimal object-id and the show snmp mib (get get-next walk) object-id commands display the same output.</p> <p>object-id—The object can be represented by a sequence of dotted integers (such as 1.3.6.1.2.1.2) or by its subtree name (such as interfaces). When entering multiple objects, enclose the objects in quotation marks.</p>
Required Privilege Level	snmp—To view this statement in the configuration.
List of Sample Output	<p>show snmp mib get on page 273</p> <p>show snmp mib get (Multiple Objects) on page 273</p> <p>show snmp mib get (Layer 2 Policer) on page 273</p> <p>show snmp mib get-next on page 273</p> <p>show snmp mib get-next (Specify an OID) on page 273</p> <p>show snmp mib walk on page 273</p> <p>show snmp mib walk (QFX Series) on page 273</p> <p>show snmp mib walk decimal on page 274</p> <p>show snmp mib walk (ASCII) on page 274</p> <p>show snmp mib walk (Multiple Indices) on page 274</p> <p>show snmp mib walk decimal (Multiple Indices) on page 274</p>
Output Fields	Table 28 on page 273 describes the output fields for the show snmp mib command. Output fields are listed in the approximate order in which they appear.

Table 28: show snmp mib Output Fields

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

Sample Output

show snmp mib get

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

show snmp mib get (Multiple Objects)

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

show snmp mib get (Layer 2 Policer)

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

show snmp mib get-next

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

show snmp mib get-next (Specify an OID)

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

show snmp mib walk

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

show snmp mib walk (QFX Series)

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

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

show snmp mib walk decimal

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

show snmp mib walk (ASCII)

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

show snmp mib walk (Multiple Indices)

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

show snmp mib walk decimal (Multiple Indices)

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

show system uptime

List of Syntax	Syntax on page 275 Syntax (EX Series Switches) on page 275 Syntax (QFX Series) on page 275 Syntax (TX Matrix Router) on page 275 Syntax (TX Matrix Plus Router) on page 275 Syntax (MX Series Router) on page 275
Syntax	show system uptime
Syntax (EX Series Switches)	show system uptime <all-members> <local> <member <i>member-id</i> >
Syntax (QFX Series)	show system uptime <director-group <i>name</i> > <infrastructure <i>name</i> > <interconnect-device <i>name</i> > <node-group <i>name</i> >
Syntax (TX Matrix Router)	show system uptime <all-chassis all-lcc lcc <i>number</i> scc>
Syntax (TX Matrix Plus Router)	show system uptime <detail> <all-chassis all-lcc lcc <i>number</i> sfc <i>number</i> >
Syntax (MX Series Router)	show system uptime <all-members> <invoke-on> <local> <member <i>member-id</i> >
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. sfc option introduced for the TX Matrix Plus router in JUNOS Release 9.6. Command introduced in Junos OS Release 11.1 for the QFX Series. Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	Display the current time and information about how long the router or switch, router or switch software, and routing protocols have been running.
Options	none —Show time since the system rebooted and processes started. all-chassis —(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. all-lcc —(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.

all-members—(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.

director-group *name*—(QFabric systems only) (Optional) Show time since the system rebooted and processes started on the Director group.

infrastructure *name*—(QFabric systems only) (Optional) Show time since the system rebooted and processes started on the fabric control Routing Engine and fabric manager Routing Engine.

interconnect-device *name*—(QFabric systems only) (Optional) Show time since the system rebooted and processes started on the Interconnect device.

invoke-on—(MX Series routers only) (Optional) Display the time since the system rebooted and processes started on the master Routing Engine, backup Routing Engine, or both, on a router with two Routing Engines.

lcc *number*—(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.

Replace *number* with the following values depending on the LCC configuration:

- 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.
- 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.
- 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.
- 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.

local—(EX4200 switches and MX Series routers only) (Optional) Show time since the system rebooted and processes started on the local Virtual Chassis member.

member *member-id*—(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 *member-id* with a value from 0 through 9. For an MX Series Virtual Chassis, replace *member-id* with a value of 0 or 1.

node-group *name*—(QFabric systems only) (Optional) Show time since the system rebooted and processes started on the Node group.

scc—(TX Matrix routers only) (Optional) Show time since the system rebooted and processes started for the TX Matrix router (or switch-card chassis).

sfc *number*—(TX Matrix Plus routers only) (Optional) Show time since the system rebooted and processes started for the TX Matrix Plus router. Replace *number* with 0.

Additional Information By default, when you issue the **show system uptime** command on the master Routing Engine of a TX Matrix router or a TX Matrix Plus router, the command is broadcast to all the master 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

Related Documentation

- [Monitoring System Process Information](#)
- [Monitoring System Properties](#)
- [10-Gigabit Ethernet LAN/WAN PIC with XFP \(T640 Router\)](#)
- [Routing Matrix with a TX Matrix Plus Router Solutions Page](#)

List of Sample Output

- [show system uptime on page 278](#)
- [show system uptime all-lcc \(TX Matrix Router\) on page 278](#)
- [show system uptime all-lcc \(TX Matrix Plus Router\) on page 278](#)
- [show system uptime \(EX Series\) on page 279](#)
- [show system uptime \(QFX Series\) on page 279](#)

Output Fields [Table 29 on page 277](#) describes the output fields for the **show system uptime** command. Output fields are listed in the approximate order in which they appear.

Table 29: show system uptime Output Fields

Field Name	Field Description
Current time	Current system time in UTC.
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.
time 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:      1998-10-13 19:45:47 UTC
System booted:     1998-10-12 20:51:41 UTC (22:54:06 ago)
Protocols started: 1998-10-13 19:33:45 UTC (00:12:02 ago)
Last configured:   1998-10-13 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 all-lcc (TX Matrix Router)

```
user@host> show system uptime all-lcc
lcc0-re0:
-----
Current time: 2004-09-13 09:55:35 PDT
System booted: 2004-09-13 03:13:55 PDT (06:41:40 ago)
Last configured: 2004-09-13 03:17:48 PDT (06:37:47 ago) by root
9:55AM PDT up 6:42, 1 user, load averages: 0.02, 0.03, 0.00
lcc2-re0:
-----
Current time: 2004-09-13 09:55:35 PDT
System booted: 2004-09-12 03:23:43 PDT (1d 06:31 ago)
Last configured: 2004-09-13 03:05:36 PDT (06:49:59 ago) by root
9:55AM PDT up 1 day, 6:32, 1 user, load averages: 0.02, 0.01, 0.00
```

show system uptime all-lcc (TX Matrix Plus Router)

```
user@host> show system uptime all-lcc
sfc0-re0:
-----
Current time: 2009-05-25 00:24:30 PDT
System booted: 2009-05-24 06:39:33 PDT (17:44:57 ago)
Protocols started: 2009-05-24 06:40:30 PDT (17:44:00 ago)
Last configured: 2009-05-24 06:33:27 PDT (17:51:03 ago) by gregdo
12:24AM up 17:45, 2 users, load averages: 0.07, 0.05, 0.01

lcc0-re0:
-----
Current time: 2009-05-25 00:24:30 PDT
System booted: 2009-05-24 06:39:46 PDT (17:44:44 ago)
error: the routing subsystem is not running
Last configured: 2009-05-24 06:40:47 PDT (17:43:43 ago) by root
12:24AM up 17:45, 0 users, load averages: 0.00, 0.00, 0.00

lcc1-re0:
-----
Current time: 2009-05-25 00:24:30 PDT
System booted: 2009-05-24 06:39:38 PDT (17:44:52 ago)
error: the routing subsystem is not running
Last configured: 2009-05-24 06:40:18 PDT (17:44:12 ago) by root
12:24AM up 17:45, 0 users, load averages: 0.00, 0.00, 0.00

lcc2-re0:
-----
Current time: 2009-05-25 00:24:30 PDT
System booted: 2009-05-24 06:39:48 PDT (17:44:42 ago)
error: the routing subsystem is not running
Last configured: 2009-05-24 06:40:44 PDT (17:43:46 ago) by root
12:24AM up 17:45, 0 users, load averages: 0.00, 0.00, 0.00
```

lcc3-re0:

```
-----  
Current time: 2009-05-25 00:24:30 PDT  
System booted: 2009-05-24 06:39:44 PDT (17:44:46 ago)  
error: the routing subsystem is not running  
Last configured: 2009-05-24 06:40:08 PDT (17:44:22 ago) by root  
12:24AM up 17:45, 0 users, load averages: 0.00, 0.00, 0.00
```

show system uptime (EX Series)

```
user@switch> show system uptime  
Current time: 2014-03-12 16:39:56 UTC  
System booted: 2014-03-12 14:58:05 UTC (01:41:51 ago)  
Protocols started: 2014-03-12 14:59:48 UTC (01:40:08 ago)  
Last configured: 2014-03-12 14:58:58 UTC (01:40:58 ago) by root  
4:39PM up 1:42, 4 users, load averages: 0.02, 0.02, 0.00
```

show system uptime (QFX Series)

```
user@switch> show system uptime  
Current time: 2010-08-27 03:12:30 PDT  
System booted: 2010-08-13 17:11:54 PDT (1w6d 10:00 ago)  
Protocols started: 2010-08-13 17:13:56 PDT (1w6d 09:58 ago)  
Last configured: 2010-08-26 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 virtual-chassis active-topology

Syntax	show virtual-chassis active-topology <all-members> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the active topology of the Virtual Chassis or VCF with next-hop reachability information.
Options	<p>none—Display the active topology of the member switch where the command is issued.</p> <p>all-members—(Optional) Display the active topology of all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display the active topology of the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display the active topology of the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247 • Understanding EX Series Virtual Chassis Configuration on page 28
List of Sample Output	show virtual-chassis active-topology (EX4200 Virtual Chassis) on page 281 show virtual-chassis active-topology (EX8200 Virtual Chassis) on page 281 show virtual-chassis active-topology (Virtual Chassis Fabric) on page 282
Output Fields	Table 30 on page 280 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 30: show virtual-chassis active-topology Output Fields

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

Sample Output

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

```

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

 2                      1(vcp-1)

 3                      1(vcp-1)

 4                      1(vcp-1)

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

 6                      8(vcp-0)

 7                      8(vcp-0)

 8                      8(vcp-0)

```

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

```

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

```

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

```

member1:

```

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

```

member8:

```

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

member9:

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

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

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

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

fpc1:

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

fpc2:

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

fpc3:

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

fpc4:

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

fpc5:

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

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

fpc6:

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

show virtual-chassis device-topology

Syntax	show virtual-chassis device-topology <all-members> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the device topology—the member and system IDs, the VCP numbers, and device status—for all hardware devices in the Virtual Chassis or VCF.
Options	<p>none—Display the device topology for all members of the Virtual Chassis or VCF.</p> <p>all-members—(Optional) Display the device topology for all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display the device topology for the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display the device topology for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • Understanding EX Series Virtual Chassis Port Link Aggregation on page 26 • Understanding EX8200 Virtual Chassis Topologies
Output Fields	Table 31 on page 285 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 31: show virtual-chassis device-topology Output Fields

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

Table 31: show virtual-chassis device-topology Output Fields (*continued*)

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

Sample Output

show virtual-chassis device-topology

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

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

member9:

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

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

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

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

fpc1:

Neighbor List

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

fpc2:

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

fpc3:

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

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

fpc4:

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

fpc5:

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

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

fpc6:

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

show virtual-chassis protocol adjacency

Syntax	<pre>show virtual-chassis protocol adjacency <brief detail extensive> <all-members> <local> <member member-id> <system-id></pre>
Release Information	<p>Command introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	Display the Virtual Chassis Control Protocol (VCCP) adjacency statistics in the Virtual Chassis or VCF for all hardware devices.
Options	<p>none—Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>brief detail extensive—(Optional) Display the specified level of output. Using the brief option is equivalent to entering the command with no options (the default). The detail and extensive options provide identical displays.</p> <p>all-members—(Optional) Display VCCP adjacency statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display VCCP adjacency statistics for the switch or external Routing Engine on which this command is entered.</p> <p>member member-id—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.</p> <p>system-id—(Optional) Display VCCP adjacency statistics for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • Understanding EX Series Virtual Chassis Port Link Aggregation on page 26 • Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis
List of Sample Output	<p>show virtual-chassis protocol adjacency on page 292</p> <p>show virtual-chassis protocol adjacency detail on page 293</p>
Output Fields	Table 32 on page 292 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 32: show virtual-chassis protocol adjacency Output Fields

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

Sample Output

show virtual-chassis protocol adjacency

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

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


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

show virtual-chassis protocol adjacency detail

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

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

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

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

```
member1:
```

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

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

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

```
member8:
```

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

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

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

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

```
member9:
```

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

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

show virtual-chassis protocol database

Syntax	show virtual-chassis protocol database <brief detail extensive> <all-members> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the Virtual Chassis Control Protocol (VCCP) database statistics for all hardware devices within the Virtual Chassis or VCF.
Options	<p>none—Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>brief detail extensive—(Optional) Display the specified level of output. Using the brief option is equivalent to entering the command with no options (the default). The detail option provides more output than the brief option. The extensive option provides all output and is most useful for customer support personnel.</p> <p>all-members—(Optional) Display VCCP database statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display VCCP database statistics for the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display VCCP database statistics for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis • Understanding EX Series Virtual Chassis Components on page 9 • Understanding QFX Series Virtual Chassis Components
List of Sample Output	show virtual-chassis protocol database on page 296 show virtual-chassis protocol database detail on page 297
Output Fields	Table 33 on page 295 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 33: show virtual-chassis protocol database Output Fields

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

Table 33: show virtual-chassis protocol database Output Fields (*continued*)

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

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

Sample Output

show virtual-chassis protocol database

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

```
member0:
```

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

```
member1:
```

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

```
member8:
```

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

```
member9:
```

```
-----
LSP ID          Sequence Checksum Lifetime
0000.4a75.9b7c.00-00  0x1dd80  0xc2e3   116
0000.73e9.9a57.00-00  0xf361   0x27e8   116
0021.59f7.d000.00-00  0x16883  0x289    114
```

```
0026.888d.6800.00-00      0x16920   0xa335      116
4 LSPs
```

show virtual-chassis protocol database detail

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

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

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

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

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

```
member1:
```

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

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

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

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

```
member8:
```

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

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

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

```
Neighbor: 0000.4a75.9b7c.00 Interface: vcp-0/0.32768 Metric: 150
Neighbor: 0026.888d.6800.00 Interface: vcp-4/0/1.32768 Metric: 15

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

member9:

```
-----

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

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

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

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

show virtual-chassis protocol interface

Syntax	<pre>show virtual-chassis protocol interface <brief detail> <all-members> <interface-name> <local> <member member-id></pre>
Release Information	<p>Command introduced in Junos OS Release 10.4 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	Display information about Virtual Chassis Control Protocol (VCCP) statistics for VCCP-enabled interfaces within the Virtual Chassis or VCF.
Options	<p>none—Display the VCCP interface statistics in brief form for all members of the Virtual Chassis or VCF.</p> <p>brief detail —(Optional) Display the specified level of output. Using the brief option is equivalent to entering the command with no options (the default). The detail option provides more output than the brief option.</p> <p>all-members—(Optional) Display VCCP interface statistics for all members of the Virtual Chassis or VCF.</p> <p>interface-name—(Optional) Display VCCP interface statistics for the specified interface.</p> <p>local—(Optional) Display VCCP interface statistics for the switch or external Routing Engine on which this command is entered.</p> <p>member member-id—(Optional) Display VCCP interface statistics for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • EX Series Virtual Chassis Overview on page 3 • 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
List of Sample Output	show virtual-chassis protocol interface on page 300
Output Fields	Table 34 on page 300 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 34: show virtual-chassis protocol interface Output Fields

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

Sample Output

show virtual-chassis protocol interface

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

```
member0:
```

```
-----
```

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

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

```
member1:
```

```
-----
```

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

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

```
member8:
```

```
-----
```

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

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

```
member9:
```

```
-----
```

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

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

show virtual-chassis protocol route

Syntax	show virtual-chassis protocol route <all-members> <destination-id> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the unicast and multicast Virtual Chassis Control Protocol (VCCP) routing tables within the Virtual Chassis or VCF.
Options	<p>none—Display the unicast and multicast routing tables for all members of the Virtual Chassis.</p> <p>all-members—(Optional) Display the unicast and multicast routing tables for all members of the Virtual Chassis or VCF.</p> <p>destination-id—(Optional) Display the unicast and multicast routing tables to the specified destination member ID for each member of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display the unicast and multicast routing tables on the device where this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display the unicast and multicast routing tables for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • EX Series Virtual Chassis Overview on page 3 • Understanding QFX Series Virtual Chassis • Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis
List of Sample Output	show virtual-chassis protocol route on page 303
Output Fields	Table 35 on page 302 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 35: show virtual-chassis protocol route Output Fields

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

Table 35: show virtual-chassis protocol route Output Fields (*continued*)

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

Sample Output

show virtual-chassis protocol route

```

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

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

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

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

```

vcp-0/1.32768

member8:

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

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

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

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

member9:

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

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

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

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

show virtual-chassis protocol statistics

Syntax	show virtual-chassis protocol statistics <all-members> <interface-name> <local> <member member-id>
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the Virtual Chassis Control Protocol (VCCP) statistics for all hardware devices within the Virtual Chassis or VCF.
Options	<p>none—Display VCCP statistics for all members of the Virtual Chassis or VCF.</p> <p>all-members—(Optional) Display VCCP statistics for all members of the Virtual Chassis or VCF.</p> <p>interface-name—(Optional) Display VCCP statistics for the specified interface.</p> <p>local—(Optional) Display VCCP statistics for the switch or external Routing Engine on which this command is entered.</p> <p>member member-id—(Optional) Display VCCP statistics for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • EX Series Virtual Chassis Overview on page 3 • Understanding QFX Series Virtual Chassis • Understanding the Virtual Chassis Control Protocol in an EX8200 Virtual Chassis
List of Sample Output	show virtual-chassis protocol statistics on page 306
Output Fields	Table 36 on page 305 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 36: show virtual-chassis protocol statistics Output Fields

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

Table 36: show virtual-chassis protocol statistics Output Fields (*continued*)

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

Sample Output

show virtual-chassis protocol statistics

```

user@switch> show virtual-chassis protocol statistics
member0:
-----
IS-IS statistics for 0021.59f7.d000:
PDU type      Received    Processed      Drops      Sent      Rexmit
LSP            8166        8166           0         4551         0
HELLO          1659        1659           0         1693         0
CSNP             2            2             0            3         0
PSNP           1909        1909           0         2293         0
Unknown         0            0             0            0         0
Totals         11736       11736           0         8540         0

Total packets received: 11736 Sent: 8540

LSP queue length: 0 Drops: 0
SPF runs: 9
Fragments rebuilt: 1640
LSP regenerations: 1
Purges initiated: 0

member1:
-----
IS-IS statistics for 0026.888d.6800:

```

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

Total packets received: 16635 Sent: 18074

LSP queue length: 0 Drops: 0
 SPF runs: 13
 Fragments rebuilt: 1871
 LSP regenerations: 2
 Purges initiated: 0

member8:

IS-IS statistics for 0000.4a75.9b7c:

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

Total packets received: 15032 Sent: 25659

LSP queue length: 0 Drops: 0
 SPF runs: 26
 Fragments rebuilt: 2666
 LSP regenerations: 4
 Purges initiated: 0

member9:

IS-IS statistics for 0000.73e9.9a57:

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

Total packets received: 14977 Sent: 11834

LSP queue length: 0 Drops: 0
 SPF runs: 19
 Fragments rebuilt: 1510
 LSP regenerations: 6
 Purges initiated: 0

show virtual-chassis fast-failover

Syntax	show virtual-chassis fast-failover
Release Information	Command introduced in Junos OS Release 9.3 for EX Series switches.
Description	Display information about the fast failover feature in a Virtual Chassis configuration.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring Fast Failover on Uplink Module VCPs to Reroute Traffic When an EX4200 Virtual Chassis Switch or Intermember Link Fails on page 131 • Configuring Fast Failover in an EX Series Virtual Chassis on page 197 • Disabling Fast Failover in an EX Series Virtual Chassis on page 198
List of Sample Output	show virtual-chassis fast-failover on page 308
Output Fields	Table 37 on page 308 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 37: show virtual-chassis fast-failover Output Fields

Field Name	Field Description
Fast failover on dedicated VCP ports	Indicates fast failover status on dedicated VCPs.
Fast failover on XE uplink VCP ports	Indicates fast failover status on XFP uplink VCPs.
Fast failover on GE uplink VCP ports	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

```


show virtual-chassis login

Syntax	show virtual-chassis login
Release Information	<p>Command introduced in Junos OS Release 9.3 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	<p>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.</p> <p>You might need this information for tracing or troubleshooting purposes.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • request session member on page 258 • Understanding Global Management of a Virtual Chassis on page 22
List of Sample Output	<p>show virtual-chassis login (Direct Login to the Master Console Port) on page 309</p> <p>show virtual-chassis login (Backup Console Session Redirected to the Master Console Port) on page 309</p>

Sample Output

show virtual-chassis login (Direct Login to the Master Console Port)

```
user@switch> show virtual-chassis login
Current login session initiated from host 248.1.2.3
```

show virtual-chassis login (Backup Console Session Redirected to the Master Console Port)

```
user@switch> show virtual-chassis login
Current login session initiated from host backup
```

show virtual-chassis mode

Syntax	show virtual-chassis mode <all-members> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 11.1 for EX Series switches. Command introduced in Junos OS Release 13.2X51-D20 for QFX Series devices. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF). Current mode and Future mode after reboot fields introduced in Junos OS Release 13.2X51-D20.
Description	Display the Virtual Chassis or Virtual Chassis Fabric (VCF) mixed mode status.
Options	<p>none—Display the Virtual Chassis or VCF mixed mode status for the device on which the command is entered.</p> <p>all-members—(Optional) Display the Virtual Chassis or VCF mixed mode status for all member devices in the Virtual Chassis or VCF.</p> <p>local—(Optional) Display the Virtual Chassis or VCF mixed mode status for the device on which the command is entered.</p> <p>member <i>member-id</i>—(Optional) Display the Virtual Chassis or VCF mixed mode status for the specified member device..</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • request virtual-chassis mode on page 261 • <i>Verifying the Virtual Chassis Fabric Mode Settings</i> • Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches (CLI Procedure) on page 165
List of Sample Output	show virtual-chassis mode (EX4200) on page 311 show virtual-chassis mode (QFX5100) on page 311
Output Fields	Table 38 on page 310 lists the output fields for the show virtual-chassis mode command.

Table 38: 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 Enabled or Disabled .

Table 38: show virtual-chassis mode Output Fields (*continued*)

Field Name	Field Description
Current mode	<p>Specifies the current mixed and fabric mode settings running on the member device or devices.</p> <p>A device reboot is required to change the fabric or mixed mode. The Current mode and Future mode after reboot 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"> • Fabric with mixed devices—Fabric mode and mixed mode are enabled. • Fabric with similar devices—Fabric mode is enabled and mixed mode is disabled. • Virtual Chassis with mixed devices—Fabric mode is disabled and mixed mode is enabled. • Virtual Chassis with similar devices—Fabric mode is disabled and mixed mode is disabled.
Future mode after reboot	<p>Specifies the mixed and fabric mode settings running on the member device or devices.</p> <p>A device reboot is required to change the fabric or mixed mode. The Current mode and Future mode after reboot 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"> • Fabric with mixed devices—Fabric mode and mixed mode are enabled. • Fabric with similar devices—Fabric mode is enabled and mixed mode is disabled. • Virtual Chassis with mixed devices—Fabric mode is disabled and mixed mode is enabled. • Virtual Chassis with similar devices—Fabric mode is disabled and mixed mode is disabled.

Sample Output

show virtual-chassis mode (EX4200)

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

Sample Output

show virtual-chassis mode (QFX5100)

```
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

show virtual-chassis

Syntax	show virtual-chassis <status>
Release Information	<p>Command introduced in Junos OS Release 9.2 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF). Fabric ID, Fabric Mode, and Route Mode output fields introduced in Junos OS Release 13.2X51-D20.</p> <p>Alias-Name output field introduced in Junos OS Release 14.1X53-D10.</p>
Description	Display information about all members of the Virtual Chassis or VCF.
Options	<p>none—Display information about all Virtual Chassis or VCF member devices.</p> <p>status—Same output as for show virtual-chassis.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show virtual-chassis active-topology on page 280 • show virtual-chassis protocol adjacency on page 291 • show virtual-chassis vc-path on page 318 • Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247
List of Sample Output	<p>show virtual-chassis (EX4200 Virtual Chassis) on page 315</p> <p>show virtual-chassis (EX8200 Virtual Chassis) on page 316</p> <p>show virtual-chassis (Virtual Chassis Fabric) on page 316</p>
Output Fields	<p>Table 39 on page 313 lists the output fields for the show virtual-chassis command. Output fields are listed in the approximate order in which they appear.</p>

Table 39: show virtual-chassis Output Fields

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

Table 39: show virtual-chassis Output Fields (*continued*)

Field Name	Field Description
Virtual Chassis Mode	<p>Mode of the Virtual Chassis or VCF. This field indicates support for the Virtual Chassis feature and, if a Virtual Chassis is configured, if it is a mixed or homogenous Virtual Chassis. Values can be:</p> <ul style="list-style-type: none"> • Enabled—The platform supports the Virtual Chassis feature. If a Virtual Chassis is currently configured, this is a homogenous Virtual Chassis (all members are the same type of switch). • Disabled—The switch does not support the Virtual Chassis feature. <p>NOTE: Switches that support the Virtual Chassis feature do not display this value. Even if a Virtual Chassis is not currently configured, those switches display Enabled in this field.</p> <ul style="list-style-type: none"> • Mixed—The platform supports the Virtual Chassis feature, and is configured as a mixed mode Virtual Chassis (members consist of more than one type of switch).
Member ID	<p>Assigned member ID and FPC:</p> <ul style="list-style-type: none"> • On all EX Series Virtual Chassis except EX8200 Virtual Chassis, and on a VCF, the FPC number refers to the member ID assigned to the switch. • On EX8200 Virtual Chassis, member IDs are numbered 0 through 9. The FPC number indicates the slot number of the line card within the Virtual Chassis. The FPC number on member 0 is always 0 through 15. The FPC number on member 1 is always 16 through 31. The FPC number on member 2 is always 32 through 47; and so on for the members.
Status	<p>For a nonprovisioned configuration:</p> <ul style="list-style-type: none"> • Prsnt for a member that is currently connected to the Virtual Chassis or VCF configuration. • NotPrsnt for a member ID that has been assigned but is not currently connected. <p>For a preprovisioned configuration:</p> <ul style="list-style-type: none"> • Prsnt for a member that is specified in the preprovisioned configuration file and is currently connected to the Virtual Chassis or VCF. • Unprvsnd for a member that is interconnected with the Virtual Chassis or VCF configuration but is not specified in the preprovisioned configuration file.
Serial No	Serial number of the member device.
Alias-Name	<p>The user-configured alias of the member device.</p> <p>The Alias-Name field appears only if an alias has been configured for at least one device in the Virtual Chassis or VCF. Aliases are configured using the alias-name statement in the <code>[edit virtual-chassis aliases serial-number <i>serial-number</i>]</code> hierarchy.</p>
Model	Model number of the member device.
Mastership Priority	Mastership priority value of the member device.
Role	Role of the member device: master, backup, or linecard.

Table 39: show virtual-chassis Output Fields (*continued*)

Field Name	Field Description
Mixed Mode	Mixed mode configuration status: <ul style="list-style-type: none"> • Y for a member device configured in mixed mode. • N for a member device not configured in mixed mode. • NA for a member device that cannot be configured in mixed mode.
Route Mode	The route mode of the member device: fabric (F) or Virtual Chassis (V).
Location	Location of the member device. If this field is empty, the location field was not set for the device.
Neighbor List	Member ID of the neighbor member to which this member's Virtual Chassis port (VCP) is connected.

Sample Output

show virtual-chassis (EX4200 Virtual Chassis)

```

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

```

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

```

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

```

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

show virtual-chassis (EX8200 Virtual Chassis)

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

Virtual Chassis ID: c806.0842.de51

Virtual Chassis Mode: Enabled

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

show virtual-chassis (Virtual Chassis Fabric)

user@switch> show virtual-chassis

Preprovisioned Virtual Chassis Fabric

Fabric ID: 0282.5fa0.3f08

Fabric Mode: Enabled

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

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

show virtual-chassis vc-path

Syntax	show virtual-chassis vc-path source-interface <i>interface-name</i> destination-interface <i>interface-name</i>
Release Information	Command introduced in Junos OS Release 9.6 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.
Description	<p>Show the forwarding path a packet takes when going from a source interface to a destination interface in a Virtual Chassis or VCF configuration.</p> <p>Starting in Junos OS Releases 14.1X53-D40, when used in VCF configurations, this command displays additional information about next hops, including when the forwarding path has multiple possible next hops within the VCF.</p>
Options	<p>source-interface <i>interface-name</i>—Name of the interface from which the packet originates in the Virtual Chassis or VCF</p> <p>destination-interface <i>interface-name</i>—Name of the interface to which the packet is being delivered in the Virtual Chassis or VCF</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247• Understanding EX Series Virtual Chassis Configuration on page 28• EX8200 Virtual Chassis Overview
List of Sample Output	<p>show virtual-chassis vc-path source-interface destination-interface (Virtual Chassis) on page 319</p> <p>show virtual-chassis vc-path source-interface destination-interface (Virtual Chassis Fabric) on page 320</p>
Output Fields	<p>Table 40 on page 319 lists the output fields for the show virtual-chassis vc-path command. Output fields are listed in the approximate order in which they appear.</p> <p>Some output field names and display order differ between the output for a Virtual Chassis and for a VCF. When this command is used to display forwarding paths in a VCF, additional fields are included in the output to show details of multiple possible next hops. The differences are described in Table 40 on page 319 and shown in sample output for each mode.</p>

Table 40: 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)	The Virtual Chassis or VCF member ID of the switch that contains the Packet Forwarding Engine for each hop through which the packet passes.
Member-ID (VCF)	
PFE-Device (Virtual Chassis)	The number of the Packet Forwarding Engine in each Virtual Chassis or VCF member through which a packet passes.
PFE (VCF)	<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.

Sample Output

show virtual-chassis vc-path source-interface destination-interface (Virtual Chassis)

```
user@switch> show virtual-chassis vc-path source-interface ge-0/0/0 destination-interface ge-1/0/1
```

```
vc-path from ge-0/0/0 to ge-1/0/1
Hop      Member  PFE-Device  Interface
0        0        1           ge-0/0/0
1        0        0           internal-1/24
2        1        3           vcp-0
3        1        4           ge-1/0/1
```

show virtual-chassis vc-path source-interface destination-interface (Virtual Chassis Fabric)

This example shows **vc-path** command output for two source-to-destination paths in the following Virtual Chassis Fabric displayed by the **show virtual-chassis** command:

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

```
Preprovisioned Virtual Chassis Fabric
Fabric ID: ec56.0915.8595
Fabric Mode: Enabled
```

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor List ID	Interface
1 (FPC 1)	Prsnt	VX1234560001	qfx5100-24q-aa	129	Backup	N	F	7	vcp-255/0/7
								8	vcp-255/0/8
								12	vcp-255/0/12
								3	vcp-255/0/3
								4	vcp-255/0/4
								5	vcp-255/0/5
								6	vcp-255/0/6
								10	vcp-255/0/10
								11	vcp-255/0/11
								9	vcp-255/0/9
2 (FPC 2)	Prsnt	VX1234560002	qfx5100-24q-aa	129	Master*	N	F	7	vcp-255/0/7
								8	vcp-255/0/8
								12	vcp-255/0/12
								3	vcp-255/0/3
								4	vcp-255/0/4
								5	vcp-255/0/5
								6	vcp-255/0/6
								10	vcp-255/0/10
								11	vcp-255/0/11
								9	vcp-255/0/9
3 (FPC 3)	Prsnt	VX1234560003	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
4 (FPC 4)	Prsnt	VX1234560004	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
5 (FPC 5)	Prsnt	VX1234560005	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
6 (FPC 6)	Prsnt	VX1234560006	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
7 (FPC 7)	Prsnt	VX1234560007	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
8 (FPC 8)	Prsnt	VX1234560008	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
9 (FPC 9)	Prsnt	VX1234560009	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
10 (FPC 10)	Prsnt	VX1234560010	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
11 (FPC 11)	Prsnt	VX1234560011	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2
12 (FPC 12)	Prsnt	VX1234560012	qfx5100-24q-aa	0	Linecard	N	F	1	vcp-255/0/1
								2	vcp-255/0/2

```
user@switch> show virtual-chassis vc-path source-interface xe-12/0/25 destination-interface xe-5/0/25
```

```
Fabric forwarding path from xe-12/0/25 (PFE 12) to xe-5/0/25 (PFE 5)
```

```
Hop 0 Member-ID 12 PFE 12
  Next-hop PFE 1
    Interface vcp-255/0/1.32768 Bandwidth 40
```

```
Next-hop PFE 2
  Interface vcp-255/0/2.32768 Bandwidth 40
Hop 1 Member-ID 1 PFE 1
  Next-hop PFE 5
    Interface vcp-255/0/5.32768 Bandwidth 40
Hop 1 Member-ID 2 PFE 2
  Next-hop PFE 5
    Interface vcp-255/0/5.32768 Bandwidth 40
Hop 2 Member-ID 5 PFE 5
```

```
user@switch> show virtual-chassis vc-path source-interface xe-12/0/25 destination-interface xe-1/0/25
```

```
Fabric forwarding path from xe-12/0/25 (PFE 12) to xe-1/0/25 (PFE 1)
Hop 0 Member-ID 12 PFE 12
  Next-hop PFE 1
    Interface vcp-255/0/1.32768 Bandwidth 40
Hop 1 Member-ID 1 PFE 1
```

show virtual-chassis vc-port

Syntax	show virtual-chassis vc-port <all-members> <local> <member <i>member-id</i> >
Release Information	Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series. Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).
Description	Display the status of the Virtual Chassis ports (VCPs), including both the dedicated VCPs and the uplink ports configured as VCPs.
Options	<p>none—Display the operational status of all VCPs of the member switch where the command is issued.</p> <p>all-members—(Optional) Display the operational status of all VCPs on all members of the Virtual Chassis or VCF.</p> <p>local—(Optional) Display the operational status of the switch or external Routing Engine on which this command is entered.</p> <p>member <i>member-id</i>—(Optional) Display the operational status of all VCPs for the specified member of the Virtual Chassis or VCF.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show virtual-chassis vc-port statistics on page 326 • Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247 • Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis
List of Sample Output	show virtual-chassis vc-port (EX4200 Virtual Chassis) on page 324 show virtual-chassis vc-port (EX8200 Virtual Chassis) on page 324 show virtual-chassis vc-port all-members on page 325
Output Fields	Table 41 on page 322 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 41: show virtual-chassis vc-port Output Fields

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

Table 41: show virtual-chassis vc-port Output Fields (*continued*)

Field Name	Field Description
Interface or PIC/Port	<p>VCP name.</p> <ul style="list-style-type: none"> The dedicated VCPs in an EX4200 or EX4500 Virtual Chassis are vcp-0 and vcp-1. The dedicated VCPs in an EX4550 Virtual Chassis are VCP-1/0, VCP-1/1, VCP-2/0, and VCP-2/1. Optical ports set as VCPs are named 1/0 and 1/1, representing the PIC number and the port number. The native VCP (port 0) on an XRE200 External Routing Engine in an EX8200 Virtual Chassis is named vcp-0. The VCPs on each Virtual Chassis Control Interface (VCCI) module in an XRE200 External Routing Engine are named using the vcp-slot-number/port-number convention; for instance, vcp-1/0. The VCPs on EX8200 member switches are named using the vcp-slot-number/pic-number/interface-number convention; for instance, vcp-3/0/2. A 255 as the first number in your port number indicates that your VCP is part of a Link Aggregation group (LAG) bundle. For instance, a display of vcp-255/1/0 indicates that the dedicated VCP named vcp-1/0 is part of a LAG bundle. A display of vcp-255/1/0 indicates that an uplink port that was previously named xe-0/1/0 is now part of a VCP LAG bundle.
Type	<p>Type of VCP:</p> <ul style="list-style-type: none"> Dedicated—The rear panel VCP on an EX4200, EX4500, or EX4550 switch, or any VCP link connected to an XRE200 External Routing Engine in an EX8200 Virtual Chassis. Configured—Optical port configured as a VCP. Auto-Configured—Optical port autoconfigured as a VCP. <p>See “Setting an Uplink Port on an EX Series Switch as a Virtual Chassis Port (CLI Procedure)” on page 188 or Setting a 10-Gigabit Ethernet Port as a Virtual Chassis Port in an EX8200 Virtual Chassis (CLI Procedure) for information about configuring VCPs.</p>
Trunk ID	<p>A positive-number ID assigned to a link aggregation group (LAG) formed by the Virtual Chassis. The trunk ID value is –1 if no trunk is formed. A LAG between uplink VCPs requires that the link speed be the same on connected interfaces and that at least two VCPs on one member be connected to at least two VCPs on the other member in an EX4200 or EX4500 Virtual Chassis.</p> <p>Dedicated VCP LAGs are assigned trunk IDs 1 and 2. Trunk IDs for LAGs formed with uplink VCPs therefore have values of 3 or greater.</p> <p>The trunk ID value changes if the link-adjacency state between LAG members changes; trunk membership is then allocated or deallocated.</p>
Status	<p>Interface status:</p> <ul style="list-style-type: none"> absent—Interface is not a VCP link. down—VCP link is down. up—VCP link is up.
Speed (mbps)	Speed of the interface in megabits per second.
Neighbor ID/Interface	The Virtual Chassis member ID and interface of a VCP on a member that is connected to the interface or PIC/Port field in the same row as this interface.

Sample Output

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

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

```
fpc0:
```

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

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

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

```
member0:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Up	1000	8	vcp-1/1
vcp-0/1	Dedicated	-1	Up	1000	8	vcp-2/0
4/0/4	Configured	-1	Up	10000	1	vcp-3/0/4
4/0/7	Configured	-1	Down	10000		
4/0/3	Configured		Absent			
4/0/2	Configured		Absent			
4/0/5	Configured		Absent			
4/0/6	Configured		Absent			
4/0/1	Configured		Absent			
4/0/0	Configured		Absent			

```
member1:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Up	1000	8	vcp-1/2
3/0/0	Configured	-1	Down	10000		
3/0/1	Configured	-1	Down	10000		
3/0/4	Configured	-1	Up	10000	0	vcp-4/0/4
3/0/5	Configured		Absent			
4/0/5	Configured		Absent			
4/0/4	Configured		Absent			

```
member8:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
vcp-0/0	Dedicated	-1	Down	1000		
vcp-1/0	Dedicated	-1	Up	1000	9	vcp-1/0
vcp-1/1	Dedicated	-1	Up	1000	0	vcp-0/0
vcp-1/2	Dedicated	-1	Up	1000	1	vcp-0/0
vcp-1/3	Dedicated	-1	Up	1000	9	vcp-1/3
vcp-2/0	Dedicated	-1	Up	1000	0	vcp-0/1
vcp-2/1	Dedicated	-1	Up	1000	9	vcp-1/2
vcp-2/2	Dedicated	-1	Down	1000		


```
vcp-2/3      Dedicated      -1   Down      1000
```

```
member9:
```

```
-----
Interface    Type            Trunk  Status    Speed    Neighbor
or           or              ID     Status    (mbps)   ID  Interface
Slot/PIC/Port
vcp-0/0      Dedicated       -1     Disabled  1000
vcp-1/0      Dedicated       -1     Up        1000      8   vcp-1/0
vcp-1/1      Dedicated       -1     Down      1000
vcp-1/2      Dedicated       -1     Up        1000      8   vcp-2/1
vcp-1/3      Dedicated       -1     Up        1000      8   vcp-1/3
```

show virtual-chassis vc-port all-members

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

```
fpc0:
```

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

```
fpc1:
```

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

```
fpc2:
```

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

show virtual-chassis vc-port statistics

Syntax	<pre>show virtual-chassis vc-port statistics <all-members> <brief detail extensive > <interface-name> <local> <member member-id></pre>
Release Information	<p>Command introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>The options all-members, brief, detail, extensive, and local were added in Junos OS Release 9.3 for EX Series switches.</p> <p>Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.</p> <p>Command introduced in Junos OS Release 13.2X51-D20 for Virtual Chassis Fabric (VCF).</p>
Description	Display the traffic statistics collected on Virtual Chassis ports (VCPs).
Options	<p>none—Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF.</p> <p>brief detail extensive—(Optional) Display the specified level of output. Using the brief option is equivalent to entering the command with no options (the default). The detail and extensive options provide identical displays.</p> <p>all-members—(Optional) Display traffic statistics for VCPs of all members of a Virtual Chassis or VCF.</p> <p>interface-name—(Optional) Display traffic statistics for the specified VCP.</p> <p>local—(Optional) Display traffic statistics for VCPs on the switch or external Routing Engine on which this command is entered.</p> <p>member member-id—(Optional) Display traffic statistics for VCPs on the specified member of a Virtual Chassis or VCF.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• clear virtual-chassis vc-port statistics on page 254• show virtual-chassis vc-port on page 322• Monitoring the Virtual Chassis Status and Statistics on EX Series Virtual Chassis on page 247• Verifying Virtual Chassis Ports in an EX8200 Virtual Chassis
List of Sample Output	<ul style="list-style-type: none">• show virtual-chassis vc-port statistics on page 329• show virtual-chassis vc-port statistics (EX8200 Virtual Chassis) on page 330• show virtual-chassis vc-port statistics brief on page 330• show virtual-chassis vc-port statistics extensive on page 330• show virtual-chassis vc-port statistics member 0 on page 332

Output Fields Table 42 on page 327 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 42: show virtual-chassis vc-port statistics Output Fields

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

Table 42: show virtual-chassis vc-port statistics Output Fields (*continued*)

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

Table 42: show virtual-chassis vc-port statistics Output Fields (*continued*)

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

Sample Output

show virtual-chassis vc-port statistics

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

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

```
{master:0}
```

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

```

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

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

member8:
-----

member9:
-----

```

show virtual-chassis vc-port statistics brief

```

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

{master:0}

```

show virtual-chassis vc-port statistics extensive

```

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

```

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

```

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

...

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

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

...

```

```
{master:0}
```

show virtual-chassis vc-port statistics member 0

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

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

```
{master:0}
```


PART 4

Troubleshooting

- [Troubleshooting Procedures on page 335](#)

CHAPTER 7

Troubleshooting Procedures

- [Troubleshooting an EX Series Virtual Chassis on page 335](#)

Troubleshooting an EX Series Virtual Chassis

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

- [A Disconnected Member Switch's ID Is Not Available for Reassignment on page 335](#)
- [Load Factory Default Does Not Commit on a Multimember Virtual Chassis on page 335](#)
- [The Member ID Persists When a Member Switch Is Disconnected From a Virtual Chassis on page 336](#)
- [A Member Switch Is Not Participating in a Mixed Virtual Chassis on page 336](#)
- [Unknown Traffic Looping Occurs After Configuring an Uplink Port as a Redundant VCP with a Dedicated VCP on page 337](#)

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

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

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

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

Load Factory Default Does Not Commit on a Multimember Virtual Chassis

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

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

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

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

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

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

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

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

1. To change the member ID to 0:

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

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

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

```
user@switch# replace pattern ge-1/ with ge-0/
```

A Member Switch Is Not Participating in a Mixed Virtual Chassis

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

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

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

To verify the Virtual Chassis mode:

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

```
fpc0:
```

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

```
fpc1:
```

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

```
fpc2:
```

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

```
fpc3:
-----
Mixed Mode: Enabled
fpc4:
-----
Mixed Mode: Disabled
fpc5:
-----
Mixed Mode: Enabled
```

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

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

(EX4500 switch only) To verify the PIC mode:

```
user@switch> show chassis pic-mode
fpc0:
-----
    Pic Mode: Not-Applicable
fpc1:
-----
    Pic Mode: Not-Applicable
fpc2:
-----
    Pic Mode: Not-Applicable
fpc3:
-----
    Pic Mode: Not-Applicable
fpc4:
-----
    Pic Mode: PIC 3: Intraconnect
fpc5:
-----
    Pic Mode: PIC 3: virtual-chassis
```

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

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

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

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

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

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.

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 on page 247](#)
- [Configuring an EX4200, EX4500, or EX4550 Virtual Chassis \(CLI Procedure\) on page 156](#)
- [Configuring a Mixed Virtual Chassis with EX4200, EX4500, and EX4550 Member Switches \(CLI Procedure\) on page 165](#)
- [Configuring a Virtual Chassis on an EX Series Switch \(J-Web Procedure\) on page 160](#)