



Junos[®] OS

MX Series Interchassis Redundancy Using Virtual Chassis

Release
13.1



Published: 2013-02-11

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

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.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Junos® OS MX Series Interchassis Redundancy Using Virtual Chassis

13.1

Copyright © 2013, Juniper Networks, Inc.
All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	xi
	Documentation and Release Notes	xi
	Supported Platforms	xi
	Using the Examples in This Manual	xi
	Merging a Full Example	xii
	Merging a Snippet	xii
	Documentation Conventions	xiii
	Documentation Feedback	xv
	Requesting Technical Support	xv
	Self-Help Online Tools and Resources	xv
	Opening a Case with JTAC	xvi
Part 1	Overview	
Chapter 1	Interchassis Redundancy and Virtual Chassis	3
	Interchassis Redundancy and Virtual Chassis Overview	3
	Interchassis Redundancy Overview	3
	Virtual Chassis Overview	4
	Supported Platforms for MX Series Virtual Chassis	4
	Benefits of Configuring a Virtual Chassis	4
	Virtual Chassis Components Overview	5
	Virtual Chassis Master Router	6
	Virtual Chassis Backup Router	6
	Virtual Chassis Line-card Router	7
	Virtual Chassis Ports	7
	Virtual Chassis Port Trunks	8
	Slot Numbering in the Virtual Chassis	8
	Virtual Chassis Control Protocol	9
	Member IDs, Roles, and Serial Numbers	9
	Guidelines for Configuring Virtual Chassis Ports	10
Chapter 2	Virtual Chassis Features	13
	Global Roles and Local Roles in a Virtual Chassis	13
	Role Name Format	13
	Global Role and Local Role Descriptions	14
	Mastership Election in a Virtual Chassis	15
	Switchover Behavior in a Virtual Chassis	17
	Virtual Chassis Role Transitions During a Global Switchover	17
	Virtual Chassis Role Transitions During a Local Switchover	18
	GRES Readiness in a Virtual Chassis Configuration	19

	Split Detection Behavior in a Virtual Chassis	19
	How Split Detection Works in a Virtual Chassis	20
	Effect of Split Detection on Virtual Chassis Failure Scenarios	20
Chapter 3	Virtual Chassis Ports and Class of Service	25
	Class of Service Overview for Virtual Chassis Ports	25
	Default CoS Configuration for Virtual Chassis Ports	25
	Supported Platforms and Maximums for CoS Configuration of Virtual Chassis Ports	26
	Default Classifiers for Virtual Chassis Ports	27
	Default Rewrite Rules for Virtual Chassis Ports	27
	Default Scheduler Map for Virtual Chassis Ports	28
	Customized CoS Configuration for Virtual Chassis Ports	29
	Output Traffic-Control Profiles	29
	Classifiers and Rewrite Rules	29
	Per-Priority Shaping	29
	Guidelines for Configuring Class of Service for Virtual Chassis Ports	30
Part 2	Configuration	
Chapter 4	Configuration Overview	33
	Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis	33
	Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers	34
	Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers	35
Chapter 5	Configuration Tasks	37
	Preparing for a Virtual Chassis Configuration	38
	Installing Junos OS Licenses on Virtual Chassis Member Routers	40
	Creating and Applying Configuration Groups for a Virtual Chassis	42
	Configuring Preprovisioned Member Information for a Virtual Chassis	44
	Enabling Graceful Routing Engine Switchover and Nonstop Active Routing for a Virtual Chassis	46
	Configuring Member IDs for a Virtual Chassis	47
	Configuring Virtual Chassis Ports to Interconnect Member Routers	49
	Deleting Virtual Chassis Ports in a Virtual Chassis Configuration	51
	Deleting Member IDs in a Virtual Chassis Configuration	52
	Switching the Global Master and Backup Roles in a Virtual Chassis Configuration	54
	Determining GRES Readiness in a Virtual Chassis Configuration	55
	Disabling Split Detection in a Virtual Chassis Configuration	57
Chapter 6	Examples	59
	Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis	59
	Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers	73

	Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers	85
	Example: Replacing a Routing Engine in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers	90
	Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers	99
Chapter 7	Configuration Statements	105
	member (MX Series Virtual Chassis)	105
	no-split-detection (MX Series Virtual Chassis)	106
	preprovisioned (MX Series Virtual Chassis)	107
	role (MX Series Virtual Chassis)	108
	serial-number (MX Series Virtual Chassis)	109
	traceoptions (MX Series Virtual Chassis)	110
	virtual-chassis (MX Series Virtual Chassis)	112
Part 3	Administration	
Chapter 8	Administrative Tasks	115
	Accessing the Virtual Chassis Through the Management Interface	116
	Managing Files on Virtual Chassis Member Routers	117
	Command Forwarding in a Virtual Chassis	118
	Virtual Chassis Slot Number Mapping for Use with SNMP	126
Chapter 9	Monitoring Tasks	129
	Verifying the Status of Virtual Chassis Member Routers	129
	Verifying the Operation of Virtual Chassis Ports	129
	Verifying Neighbor Reachability for Member Routers in a Virtual Chassis	130
	Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis	130
	Viewing Information in the Virtual Chassis Control Protocol Adjacency Database	131
	Viewing Information in the Virtual Chassis Control Protocol Link-State Database	131
	Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database	132
	Viewing Virtual Chassis Control Protocol Routing Tables	132
	Viewing Virtual Chassis Control Protocol Statistics for Member Routers and Virtual Chassis Ports	133
Chapter 10	Administrative Commands	135
	request virtual-chassis member-id delete (MX Series Virtual Chassis)	136
	request virtual-chassis member-id set (MX Series Virtual Chassis)	137
	request virtual-chassis routing-engine master switch (MX Series Virtual Chassis)	138
	request virtual-chassis vc-port delete (MX Series Virtual Chassis)	140
	request virtual-chassis vc-port set (MX Series Virtual Chassis)	142
Chapter 11	Monitoring Commands	145
	show virtual-chassis active-topology (MX Series Virtual Chassis)	146
	show virtual-chassis device-topology (MX Series Virtual Chassis)	148
	show virtual-chassis protocol adjacency (MX Series Virtual Chassis)	150

	show virtual-chassis protocol database (MX Series Virtual Chassis)	154
	show virtual-chassis protocol interface (MX Series Virtual Chassis)	160
	show virtual-chassis protocol route (MX Series Virtual Chassis)	163
	show virtual-chassis protocol statistics (MX Series Virtual Chassis)	166
	show virtual-chassis status (MX Series Virtual Chassis)	170
	show virtual-chassis vc-port (MX Series Virtual Chassis)	172
Part 4	Troubleshooting	
Chapter 12	Acquiring Troubleshooting Information	177
	Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers . . .	178
	Configuring the Name of the Virtual Chassis Trace Log File	179
	Configuring Characteristics of the Virtual Chassis Trace Log File	179
	Configuring Access to the Virtual Chassis Trace Log File	180
	Using Regular Expressions to Refine the Output of the Virtual Chassis Trace Log File	181
	Configuring the Virtual Chassis Operations to Trace	182
Chapter 13	Troubleshooting Configuration Statements	185
	traceoptions (MX Series Virtual Chassis)	186
Part 5	Index	
	Index	191

List of Figures

Part 1	Overview	
Chapter 1	Interchassis Redundancy and Virtual Chassis	3
	Figure 1: Sample Topology for MX Series Virtual Chassis	5
Part 2	Configuration	
Chapter 6	Examples	59
	Figure 2: Sample Topology for a Virtual Chassis with Two MX Series Routers . . .	60
	Figure 3: Sample Topology for a Virtual Chassis with Two MX Series Routers . . .	75
	Figure 4: Sample Topology for a Virtual Chassis with Two MX Series Routers . . .	86
	Figure 5: Sample Topology for a Virtual Chassis with Two MX Series Routers . . .	91

List of Tables

	About the Documentation	xi
	Table 1: Notice Icons	xiii
	Table 2: Text and Syntax Conventions	xiii
Part 1	Overview	
Chapter 2	Virtual Chassis Features	13
	Table 3: Global Roles and Local Roles in an MX Series Virtual Chassis	14
	Table 4: Virtual Chassis Role Transitions During Global Switchover	17
	Table 5: Virtual Chassis Role Transitions During Local Switchover	18
	Table 6: Effect of Split Detection on Common Virtual Chassis Failure Scenarios	21
Part 2	Configuration	
Chapter 5	Configuration Tasks	37
	Table 7: Virtual Chassis Global Role Transitions Before and After Mastership Switchover	54
Chapter 6	Examples	59
	Table 8: Components of the Sample MX Series Virtual Chassis	61
	Table 9: Components of the Sample MX Series Virtual Chassis	75
	Table 10: Components of the Sample MX Series Virtual Chassis	87
	Table 11: Components of the Sample MX Series Virtual Chassis	92
	Table 12: Virtual Chassis Role Transitions Before and After Local Routing Engine Switchover	98
	Table 13: Sample CoS Scheduler Hierarchy for Virtual Chassis Ports	100
Part 3	Administration	
Chapter 8	Administrative Tasks	115
	Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis	118
	Table 15: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series Virtual Chassis	126
Chapter 11	Monitoring Commands	145
	Table 16: show virtual-chassis active-topology Output Fields	146
	Table 17: show virtual-chassis device-topology Output Fields	148
	Table 18: show virtual-chassis protocol adjacency Output Fields	150
	Table 19: show virtual-chassis protocol database Output Fields	154
	Table 20: show virtual-chassis protocol interface Output Fields	160

	Table 21: show virtual-chassis protocol route Output Fields	163
	Table 22: show virtual-chassis protocol statistics Output Fields	166
	Table 23: show virtual-chassis status Output Fields	170
	Table 24: show virtual-chassis vc-port Output Fields	172
Part 4	Troubleshooting	
Chapter 12	Acquiring Troubleshooting Information	177
	Table 25: Tracing Flags for MX Series Virtual Chassis	182

About the Documentation

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

Documentation and Release Notes

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

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

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

Supported Platforms

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

- MX Series

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons


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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active

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

Convention	Description	Examples
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies book 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 System Basics Configuration 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)	Enclose 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)	Enclose a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identify 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.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web 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 send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release 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 JNASC 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: <https://www.juniper.net/alerts/>

- 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

- [Interchassis Redundancy and Virtual Chassis on page 3](#)
- [Virtual Chassis Features on page 13](#)
- [Virtual Chassis Ports and Class of Service on page 25](#)

CHAPTER 1

Interchassis Redundancy and Virtual Chassis

- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
- [Virtual Chassis Components Overview on page 5](#)
- [Guidelines for Configuring Virtual Chassis Ports on page 10](#)

Interchassis Redundancy and Virtual Chassis Overview

As more high-priority voice and video traffic is carried on the network, interchassis redundancy has become a baseline requirement for providing stateful redundancy on broadband subscriber management equipment such as broadband services routers, broadband network gateways, and broadband remote access servers. To provide a stateful interchassis redundancy solution for MX Series 3D Universal Edge Routers, you can configure a Virtual Chassis.

This topic provides an overview of interchassis redundancy and the Virtual Chassis, and explains the benefits of configuring a Virtual Chassis on supported MX Series routers.

- [Interchassis Redundancy Overview on page 3](#)
- [Virtual Chassis Overview on page 4](#)
- [Supported Platforms for MX Series Virtual Chassis on page 4](#)
- [Benefits of Configuring a Virtual Chassis on page 4](#)

Interchassis Redundancy Overview

Traditionally, redundancy in broadband edge equipment has used an intrachassis approach, which focuses on providing redundancy within a single system. However, a single-system redundancy mechanism no longer provides the degree of high availability required by service providers who must carry mission-critical voice and video traffic on their network. Consequently, service providers are requiring interchassis redundancy solutions that can span multiple systems that are colocated or geographically dispersed.

Interchassis redundancy is a high availability feature that prevents network outages and protects routers against access link failures, uplink failures, and wholesale chassis failures without visibly disrupting the attached subscribers or increasing the network management burden for service providers. Network outages can cause service providers to lose revenues

and require them to register formal reports with government agencies. A robust interchassis redundancy implementation enables service providers to fulfill strict service-level agreements (SLAs) and avoid unplanned network outages to better meet the needs of their customers.

Virtual Chassis Overview

One approach to providing interchassis redundancy is the Virtual Chassis model. In general terms, a *Virtual Chassis* configuration enables a collection of member routers to function as a single virtual router, and extends the features available on a single router to the member routers in the Virtual Chassis. The interconnected member routers in a Virtual Chassis are managed as a single network element that appears to the network administrator as a single chassis with additional line card slots, and to the access network as a single system.

To provide a stateful interchassis redundancy solution for MX Series 3D Universal Edge Routers, you can configure a Virtual Chassis. An MX Series Virtual Chassis interconnects two MX Series routers into a logical system that you can manage as a single network element. The member routers in a Virtual Chassis are designated as the *Virtual Chassis master router* (also known as the *protocol master*) and the *Virtual Chassis backup router* (also known as the *protocol backup*). The member routers are interconnected by means of dedicated *Virtual Chassis ports* that you configure on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces.

An MX Series Virtual Chassis is managed by the *Virtual Chassis Control Protocol (VCCP)*, which is a dedicated control protocol based on IS-IS. VCCP runs on the Virtual Chassis port interfaces and is responsible for building the Virtual Chassis topology, electing the Virtual Chassis master router, and establishing the interchassis routing table to route traffic within the Virtual Chassis.

Supported Platforms for MX Series Virtual Chassis

You can configure a Virtual Chassis on the following MX Series 3D Universal Edge Routers with Trio MPC/MIC interfaces (for configuration of Virtual Chassis ports) and dual Routing Engines:

- MX240 3D Universal Edge Router
- MX480 3D Universal Edge Router
- MX960 3D Universal Edge Router

In addition, graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) must be enabled on both member routers in the Virtual Chassis.

Benefits of Configuring a Virtual Chassis

Configuring a Virtual Chassis for MX Series routers provides the following benefits:

- Simplifies network management of two routers that are either colocated or geographically dispersed across a Layer 2 point-to-point network.
- Provides resiliency against network outages and protects member routers against access link failures, uplink failures, and chassis failures without visibly disrupting

attached subscribers or increasing the network management burden for service providers.

- Extends the high availability capabilities of applications such as graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) beyond a single MX Series router to both member routers in the Virtual Chassis.
- Enables service providers to fulfill strict service level agreements (SLAs) and avoid unplanned network outages to better meet their customers' needs.
- Provides the ability to scale bandwidth and service capacity as more high-priority voice and video traffic is carried on the network.

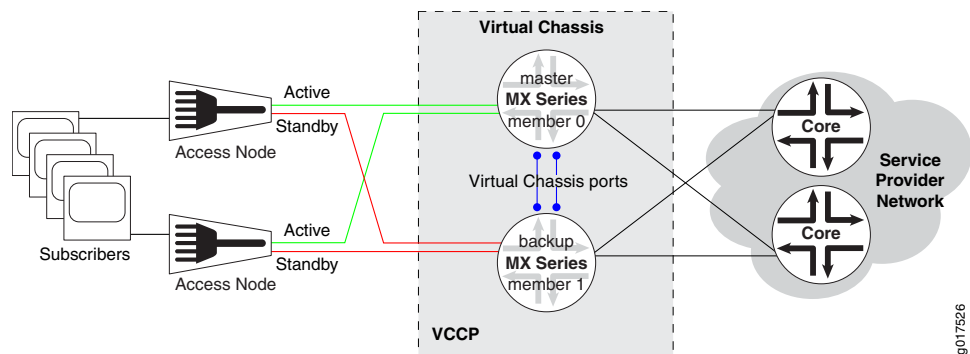
Related Documentation

- [Virtual Chassis Components Overview on page 5](#)
- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Virtual Chassis Components Overview

A Virtual Chassis configuration for MX Series 3D Universal Edge Routers interconnects two MX Series routers into a logical system that you can manage as a single network element. [Figure 1 on page 5](#) illustrates a typical topology for a two-member MX Series Virtual Chassis.

Figure 1: Sample Topology for MX Series Virtual Chassis



This overview describes the basic hardware and software components of the Virtual Chassis configuration illustrated in [Figure 1 on page 5](#), and covers the following topics:

- [Virtual Chassis Master Router on page 6](#)
- [Virtual Chassis Backup Router on page 6](#)
- [Virtual Chassis Line-card Router on page 7](#)
- [Virtual Chassis Ports on page 7](#)
- [Virtual Chassis Port Trunks on page 8](#)
- [Slot Numbering in the Virtual Chassis on page 8](#)

- [Virtual Chassis Control Protocol on page 9](#)
- [Member IDs, Roles, and Serial Numbers on page 9](#)

Virtual Chassis Master Router

One of the two member routers in the Virtual Chassis becomes the *master router*, also known as the *protocol master*. The Virtual Chassis master router maintains the global configuration and state information for both member routers, and runs the chassis management processes. The master Routing Engine that resides in the Virtual Chassis master router becomes the global master for the Virtual Chassis.

Specifically, the master Routing Engine that resides in the Virtual Chassis master router performs the following functions in a Virtual Chassis:

- Manages both the master and backup member routers
- Runs the chassis management processes and control protocols
- Receives and processes all incoming and exception path traffic destined for the Virtual Chassis
- Propagates the Virtual Chassis configuration (including member IDs, roles, and configuration group definitions and applications) to the members of the Virtual Chassis

The first member of the Virtual Chassis becomes the initial master router by default. After the Virtual Chassis is formed with both member routers, the Virtual Chassis Control Protocol (VCCP) software runs a mastership election algorithm to elect the master router for the Virtual Chassis configuration.



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

Virtual Chassis Backup Router

The member router in the Virtual Chassis that is not designated as the master router becomes the *backup router*, also known as the *protocol backup*. The Virtual Chassis backup router takes over mastership of the Virtual Chassis if the master router is unavailable, and synchronizes routing and state information with the master router. The master Routing Engine that resides in the Virtual Chassis backup router becomes the global backup for the Virtual Chassis.

Specifically, the master Routing Engine that resides in the Virtual Chassis backup router performs the following functions in a Virtual Chassis:

- If the master router fails or is unavailable, takes over mastership of the Virtual Chassis in order to preserve routing information and maintain network connectivity without disruption
- Synchronizes routing and application state, including routing tables and subscriber state information, with the master Routing Engine that resides in the Virtual Chassis master router

- Relays chassis control information, such as line card presence and alarms, to the master router

Virtual Chassis Line-card Router



NOTE: The line-card role is not supported in the preprovisioned configuration for a two-member MX Series Virtual Chassis. In this release, the line-card role applies only in the context of split detection behavior.

A member router functioning in the **line-card** role runs only a minimal set of chassis management processes required to relay chassis control information, such as line card presence and alarms, to the Virtual Chassis master router.

You cannot explicitly configure a member router with the **line-card** role in the current release. However, if the backup router fails in a two-member Virtual Chassis configuration and split detection is enabled (the default behavior), the master router takes a **line-card** role, and line cards (FPCs) that do not host Virtual Chassis ports go offline. This state effectively isolates the master router and removes it from the Virtual Chassis until connectivity is restored. As a result, routing is halted and the Virtual Chassis configuration is disabled.

Virtual Chassis Ports

Virtual Chassis ports are special Ethernet interfaces that form a point-to-point connection between the member routers in a Virtual Chassis. When you create a Virtual Chassis, you must configure the Virtual Chassis ports on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces. After you configure a Virtual Chassis port, it is renamed **vcp-slot/pic/port** (for example, **vcp-2/2/0**), and the line card associated with that port comes online. For example, the sample Virtual Chassis topology shown in [Figure 1 on page 5](#) has a total of four Virtual Chassis ports (represented by the blue dots), two on each of the two member routers.

After a Virtual Chassis port is configured, it is dedicated to the task of interconnecting member routers, and is no longer available for configuration as a standard network port. To restore this port to the global configuration and make it available to function as a standard network port, you must delete the Virtual Chassis port from the Virtual Chassis configuration.

You can configure a Virtual Chassis port on either a 1-Gigabit Ethernet (**ge**) interface or a 10-Gigabit Ethernet (**xe**) interface. However, you cannot configure a combination of 1-Gigabit Ethernet Virtual Chassis ports and 10-Gigabit Ethernet Virtual Chassis ports in the same Virtual Chassis. We recommend that you configure Virtual Chassis ports only on 10-Gigabit Ethernet interfaces. In addition, to minimize network disruption in the event of a router or link failure, configure redundant Virtual Chassis ports that reside on different line cards in each member router.

Virtual Chassis port interfaces carry both VCCP packets and internal control and data traffic. Because the internal control traffic is neither encrypted nor authenticated, make

sure the Virtual Chassis port interfaces are properly secured to prevent malicious third-party attacks on the data.

Virtual Chassis ports use a default class of service (CoS) configuration that applies equally to all Virtual Chassis port interfaces configured in a Virtual Chassis. Optionally, you can create a customized CoS traffic-control profile and apply it to all Virtual Chassis port interfaces. For example, you might want to create a nondefault traffic-control profile that allocates more than the default 5 percent of the Virtual Chassis port bandwidth to control traffic, or that assigns different priorities and excess rates to different forwarding classes.

Virtual Chassis Port Trunks

If two or more Virtual Chassis ports of the same type and speed are configured between the same two member routers in an MX Series Virtual Chassis, the Virtual Chassis Control Protocol (VCCP) bundles these Virtual Chassis port interfaces into a trunk, reduces the routing cost accordingly, and performs traffic load balancing across all of the Virtual Chassis port interfaces (also referred to as Virtual Chassis port links) in the trunk.

A Virtual Chassis port trunk must include only Virtual Chassis ports of the same type and speed. For example, a Virtual Chassis port trunk can include either all 10-Gigabit Ethernet (**xe** media type) Virtual Chassis ports or all 1-Gigabit Ethernet (**ge** media type) Virtual Chassis ports. An MX Series Virtual Chassis does *not* support a combination of 1-Gigabit Ethernet Virtual Chassis ports and 10-Gigabit Ethernet Virtual Chassis ports in the same Virtual Chassis port trunk.

The router uses the following formula to determine the cost metric of a Virtual Chassis port link in a Virtual Chassis port trunk:

$$\text{Cost} = (300 * 1,000,000,000) / \text{port-speed}$$

where *port-speed* is the aggregate speed, in bits per second, of the Virtual Chassis port.

For example, a 10-Gigabit Ethernet Virtual Chassis port link has a cost metric of 30 ($300 * 1,000,000,000 / 10,000,000,000$). A 1-Gigabit Ethernet Virtual Chassis port link has a cost metric of 300 ($300 * 1,000,000,000 / 1,000,000,000$). Virtual Chassis port links with a lower cost metric are preferred over those with a higher cost metric.

An MX Series Virtual Chassis supports up to 16 Virtual Chassis ports per trunk.

Slot Numbering in the Virtual Chassis

When the Virtual Chassis forms, the slots for line cards (FPCs) that do not host Virtual Chassis ports are renumbered to reflect the slot numbering and offsets used in the Virtual Chassis instead of the physical slot numbers where the line card is actually installed. In a two-member MX Series Virtual Chassis, member 0 in the Virtual Chassis uses FPC slot numbers 0 through 11 with no offset, and member 1 uses FPC slot numbers 12 through 23, with an offset of 12.

For example, a 10-Gigabit Ethernet interface that appears as **xe-14/2/2** (FPC slot 14, PIC slot 2, port 2) in the **show interfaces** command output is actually physical interface

xe-2/2/2 (FPC slot 2, PIC slot 2, port 2) on member 1 after deducting the FPC slot numbering offset of 12 for member 1.

The slot numbering for Virtual Chassis ports uses the physical slot number where the Virtual Chassis port is configured. For example, **vcp-3/2/0** is configured on physical FPC slot 3, PIC slot 2, port 0.



NOTE: For information about how the slot numbering in an MX Series Virtual Chassis affects the use of SNMP, see [“Virtual Chassis Slot Number Mapping for Use with SNMP” on page 126](#).

Virtual Chassis Control Protocol

An MX Series Virtual Chassis is managed by the Virtual Chassis Control Protocol (VCCP), which is a dedicated control protocol based on IS-IS. VCCP runs on the Virtual Chassis port interfaces and performs the following functions in the Virtual Chassis:

- Discovers and builds the Virtual Chassis topology
- Runs the mastership election algorithm to determine the Virtual Chassis master router
- Establishes the interchassis routing table to route traffic within the Virtual Chassis

Like IS-IS, VCCP exchanges link-state PDUs for each member router to construct a shortest path first (SPF) topology and to determine each member router's role (master or backup) in the Virtual Chassis. Because VCCP supports only point-to-point connections, no more than two member routers can be connected on any given Virtual Chassis port interface.

Member IDs, Roles, and Serial Numbers

To configure an MX Series Virtual Chassis, you must create a preprovisioned configuration that provides the following required information for each member router:

- Member ID—A numeric value (0 or 1) that identifies the member router in a Virtual Chassis configuration.
- Role—The role to be performed by each member router in the Virtual Chassis. In a two-member MX Series Virtual Chassis, you must assign both member routers the **routing-engine** role, which enables either router to function as the master router or backup router of the Virtual Chassis.
- Serial number—The chassis serial number of each member router in the Virtual Chassis. To obtain the router's serial number, find the label affixed to the side of the MX Series chassis, or issue the **show chassis hardware** command on the router to display the serial number in the command output.

The preprovisioned configuration permanently associates the member ID and role with the member router's chassis serial number. When a new member router joins the Virtual Chassis, the VCCP software compares the router's serial number against the values specified in the preprovisioned configuration. If the serial number of a joining router does

not match any of the configured serial numbers, the VCCP software prevents that router from becoming a member of the Virtual Chassis.

**Related
Documentation**

- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
- [Guidelines for Configuring Virtual Chassis Ports on page 10](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 13](#)
- [Split Detection Behavior in a Virtual Chassis on page 19](#)
- [Virtual Chassis Slot Number Mapping for Use with SNMP on page 126](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Guidelines for Configuring Virtual Chassis Ports

To interconnect the member routers in a Virtual Chassis for MX Series 3D Universal Edge Routers, you must configure Virtual Chassis ports on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces. After it is configured, a Virtual Chassis port is dedicated to the task of interconnecting member routers, and is no longer available for configuration as a standard network port.

Consider the following guidelines when you configure Virtual Chassis ports in an MX Series Virtual Chassis:

- An MX Series Virtual Chassis supports up to 16 Virtual Chassis ports per trunk.

If two or more Virtual Chassis ports of the same type and speed (that is, either all 10-Gigabit Ethernet Virtual Chassis ports or all 1-Gigabit Ethernet Virtual Chassis ports) are configured between the same two member routers in an MX Series Virtual Chassis, the Virtual Chassis Control Protocol (VCCP) bundles these Virtual Chassis port interfaces into a trunk, reduces the routing cost accordingly, and performs traffic load balancing across all of the Virtual Chassis port interfaces in the trunk.

- An MX Series Virtual Chassis does *not* support a combination of 1-Gigabit Ethernet (**ge** media type) Virtual Chassis ports and 10-Gigabit Ethernet (**xe** media type) Virtual Chassis ports within the same Virtual Chassis.

You must configure either all 10-Gigabit Virtual Chassis ports or all 1-Gigabit Virtual Chassis ports in the same Virtual Chassis. We recommend that you configure Virtual Chassis ports on 10-Gigabit Ethernet (**xe**) interfaces.

This restriction has no effect on access ports or uplink ports in an MX Series Virtual Chassis configuration.

- Configure redundant Virtual Chassis ports that reside on different line cards in each member router.

For a two-member MX Series Virtual Chassis, we recommend that you configure a minimum of two 10-Gigabit Ethernet Virtual Chassis ports on different line cards in each member router, for a total minimum of four 10-Gigabit Ethernet Virtual Chassis ports in the Virtual Chassis. In addition, make sure the Virtual Chassis port bandwidth

is equivalent to no less than 50 percent of the aggregate bandwidth required for user data traffic. The following examples illustrate these recommendations:

- If the bandwidth in your network is equivalent to two 10-Gigabit Ethernet interfaces (20 Gbps) on the access-facing side of the Virtual Chassis and two 10-Gigabit Ethernet interfaces (20 Gbps) on the core-facing side of the Virtual Chassis, we recommend that you configure two 10-Gigabit Ethernet Virtual Chassis ports, which is the recommended minimum in a Virtual Chassis for redundancy purposes.
- If the aggregate bandwidth in your network is equivalent to ten 10-Gigabit Ethernet interfaces (100 Gbps), we recommend that you configure a minimum of five 10-Gigabit Ethernet Virtual Chassis ports, which is 50 percent of the aggregate bandwidth.
- A user data packet traversing the Virtual Chassis port interfaces between member routers is discarded at the Virtual Chassis egress port if the MTU size of the packet exceeds 9150 bytes.

The maximum MTU size of a Gigabit Ethernet interface or 10-Gigabit Ethernet interface on a single MX Series router is 9192 bytes. In an MX Series Virtual Chassis configuration, user data packets that traverse Gigabit Ethernet or 10-Gigabit Ethernet Virtual Chassis port interfaces have 42 extra bytes of Virtual Chassis-specific header data, which reduces their maximum MTU (payload) size to 9150 bytes. The user data packet is transmitted in its entirety across the Virtual Chassis port interface. However, because packet fragmentation and reassembly is not supported on Virtual Chassis port interfaces, user data packets that exceed 9150 bytes are discarded at the Virtual Chassis egress port.

**Related
Documentation**

- [Virtual Chassis Components Overview on page 5](#)
- [Configuring Virtual Chassis Ports to Interconnect Member Routers on page 49](#)
- [Class of Service Overview for Virtual Chassis Ports on page 25](#)
- [Guidelines for Configuring Class of Service for Virtual Chassis Ports on page 30](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

CHAPTER 2

Virtual Chassis Features

- [Global Roles and Local Roles in a Virtual Chassis on page 13](#)
- [Mastership Election in a Virtual Chassis on page 15](#)
- [Switchover Behavior in a Virtual Chassis on page 17](#)
- [Split Detection Behavior in a Virtual Chassis on page 19](#)

Global Roles and Local Roles in a Virtual Chassis

In a Virtual Chassis configuration for MX Series 3D Universal Edge Routers, each of the two member routers and each of the two Routing Engines in each member router has a distinct role. A *global role* defines the function of each member router in the Virtual Chassis, and applies globally across the entire Virtual Chassis. A *local role* defines the function of each Routing Engine in the member router, and applies locally only to that member router.

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

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

- [Role Name Format on page 13](#)
- [Global Role and Local Role Descriptions on page 14](#)

Role Name Format

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

VC-*GlobalRole*<*LocalRole*>

where:

- ***GlobalRole*** applies to the global function of the member router for the entire Virtual Chassis, and can be one of the following:

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

Global Role and Local Role Descriptions

Table 3 on page 14 describes the global roles and local roles in an MX Series Virtual Chassis.

Table 3: Global Roles and Local Roles in an MX Series Virtual Chassis

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

Table 3: Global Roles and Local Roles in an MX Series Virtual Chassis (*continued*)

Virtual Chassis Role	Type of Role	Description
VC-Ls	Local	Standby Routing Engine in the Virtual Chassis line-card router NOTE: The line-card role is not supported in the preprovisioned configuration for a two-member MX Series Virtual Chassis. The line-card role applies only in the context of split detection behavior.

Related Documentation

- [Virtual Chassis Components Overview on page 5](#)
- [Mastership Election in a Virtual Chassis on page 15](#)
- [Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 54](#)
- [Disabling Split Detection in a Virtual Chassis Configuration on page 57](#)

Mastership Election in a Virtual Chassis

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

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



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

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

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

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

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

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

```
user@host> show virtual-chassis protocol database member 0 extensive
```

```
...
```

```
TLVs:
```

```
Node Info: Member ID: 1, VC ID: 5a6a.e747.8511, Flags:3, Priority: 129
```

```
System ID: 001d.b510.0800, Device ID: 1
```

```
...
```

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

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

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

To display the mastership priority value for the member routers in the Virtual Chassis, issue the [show virtual-chassis status](#) command.

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



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

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

**Related
Documentation**

- [Virtual Chassis Components Overview on page 5](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 13](#)

- [Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 54](#)

Switchover Behavior in a Virtual Chassis

When an active or primary hardware or software component fails or is temporarily shut down, you can manually configure a *switchover* to a backup component that takes over the functions of the unavailable primary component. You can configure two types of switchovers in a Virtual Chassis configuration for MX Series 3D Universal Edge Routers:

- Global switchover—Changes the mastership in an MX Series Virtual Chassis by switching the global roles of the master router and backup router in the Virtual Chassis configuration.
- Local switchover—Toggles the local mastership of the dual Routing Engines in a member router of the Virtual Chassis.

During a switchover, the roles assigned to the member routers and Routing Engines in a Virtual Chassis configuration change. This topic describes the role transitions that occur so you can better understand how an MX Series Virtual Chassis behaves during a global or local switchover. The topic also describes how you can determine whether the member routers are ready for a global graceful Routing Engine switchover (GRES) operation from a database synchronization perspective.

- [Virtual Chassis Role Transitions During a Global Switchover on page 17](#)
- [Virtual Chassis Role Transitions During a Local Switchover on page 18](#)
- [GRES Readiness in a Virtual Chassis Configuration on page 19](#)

Virtual Chassis Role Transitions During a Global Switchover

To change the mastership in an MX Series Virtual Chassis and cause a global switchover, you issue the `request virtual-chassis routing-engine master switch` command from the master router. After you issue this command, the current master router in the Virtual Chassis (VC-M) becomes the backup router (VC-B), and the current backup router (VC-B) becomes the master router (VC-M).

A global switchover in an MX Series Virtual Chassis causes the role transitions listed in [Table 4 on page 17](#).

Table 4: Virtual Chassis Role Transitions During Global Switchover

Virtual Chassis Role <i>Before</i> Global Switchover	Virtual Chassis Role <i>After</i> Global Switchover
Virtual Chassis master router (VC-M)	Virtual Chassis backup router (VC-B)
Virtual Chassis backup router (VC-B)	Virtual Chassis master router (VC-M)
Master Routing Engine in the Virtual Chassis master router (VC-Mm)	Master Routing Engine in the Virtual Chassis backup router (VC-Bm)

Table 4: Virtual Chassis Role Transitions During Global Switchover (*continued*)

Virtual Chassis Role <i>Before</i> Global Switchover	Virtual Chassis Role <i>After</i> Global Switchover
Standby Routing Engine in the Virtual Chassis master router (VC-Ms)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)
Master Routing Engine in the Virtual Chassis backup router (VC-Bm)	Master Routing Engine in the Virtual Chassis master router (VC-Mm)
Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Standby Routing Engine in the Virtual Chassis master router (VC-Ms)

The local roles (**master** and **standby**, or **m** and **s**) of the Routing Engines do not change after a global switchover. For example, as shown in [Table 4 on page 17](#), the master Routing Engine in the Virtual Chassis backup router (VC-Bm) remains the master Routing Engine in the Virtual Chassis master router (VC-Mm) after the global switchover.

Virtual Chassis Role Transitions During a Local Switchover

To ensure redundancy in a two-member MX Series Virtual Chassis configuration, each of the two member routers must be configured with dual Routing Engines. To toggle local mastership between the master Routing Engine and the standby Routing Engine in the member router, you issue the **request chassis routing-engine master switch** command.

A local switchover in an MX Series Virtual Chassis causes the role transitions listed in [Table 5 on page 18](#).

Table 5: Virtual Chassis Role Transitions During Local Switchover

Virtual Chassis Role <i>Before</i> Local Switchover	Virtual Chassis Role <i>After</i> Local Switchover
Master Routing Engine in the Virtual Chassis master router (VC-Mm)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)
Standby Routing Engine in the Virtual Chassis master router (VC-Ms)	Master Routing Engine in the Virtual Chassis backup router (VC-Bm)
Master Routing Engine in the Virtual Chassis backup router (VC-Bm)	Master Routing Engine in the Virtual Chassis master router (VC-Mm)
Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Standby Routing Engine in the Virtual Chassis master router (VC-Ms)

The local roles (**master** and **standby**, or **m** and **s**) of the Routing Engines in the Virtual Chassis master router change after a local switchover, but the local roles of the Routing Engines in the Virtual Chassis backup router do not change. For example, as shown in [Table 5 on page 18](#), the master Routing Engine in the Virtual Chassis master router

(VC-Mm) becomes the standby Routing Engine in the Virtual Chassis backup router (VC-Bs) after the local switchover. By contrast, the master Routing Engine in the Virtual Chassis backup router (VC-Bm) remains the master Routing Engine in the Virtual Chassis master router (VC-Mm) after the local switchover.

GRES Readiness in a Virtual Chassis Configuration

Depending on the router configuration, a variable amount of time is required before a router is ready to perform a graceful Routing Engine switchover (GRES). Attempting a GRES operation before the router is ready can cause system errors and unexpected behavior. To determine whether the member routers in an MX Series Virtual Chassis configuration are ready for a GRES operation from a database synchronization perspective, you can issue the **request virtual-chassis routing-engine master switch check** command from the Virtual Chassis master router (VC-Mm) before you initiate the GRES operation.

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

Using the **request virtual-chassis routing-engine master switch check** command before you initiate the GRES operation ensures that the subscriber management and kernel databases on both member routers in an MX Series Virtual Chassis are synchronized and ready for the GRES operation.

Related Documentation

- [Virtual Chassis Components Overview on page 5](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 13](#)
- [Mastership Election in a Virtual Chassis on page 15](#)
- [Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 54](#)
- [Determining GRES Readiness in a Virtual Chassis Configuration on page 55](#)
- [Understanding Graceful Routing Engine Switchover in the Junos OS](#)

Split Detection Behavior in a Virtual Chassis

If there is a disruption to a Virtual Chassis configuration for MX Series 3D Universal Edge Routers due to the failure of a member router or one or more Virtual Chassis port interfaces, the resulting connectivity loss can cause a split in the Virtual Chassis configuration. *Split detection* identifies the split and can minimize further network disruption.

This topic covers:

- [How Split Detection Works in a Virtual Chassis on page 20](#)
- [Effect of Split Detection on Virtual Chassis Failure Scenarios on page 20](#)

How Split Detection Works in a Virtual Chassis

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

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



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

Effect of Split Detection on Virtual Chassis Failure Scenarios

The behavior of a Virtual Chassis during certain failure scenarios depends on whether split detection is enabled or disabled. [Table 6 on page 21](#) describes the effect of the split detection setting on common failure scenarios in a two-member MX Series Virtual Chassis.

Table 6: Effect of Split Detection on Common Virtual Chassis Failure Scenarios

Type of Failure	Split Detection Setting	Results
Virtual Chassis port interfaces go down	Enabled	<ul style="list-style-type: none"> VC-B takes VC-M role. Previous VC-M takes line-card (VC-L) role. The line-card role isolates the router and removes it from the Virtual Chassis until connectivity is restored. Result is a single-member Virtual Chassis consisting of only a single VC-M. The VC-M continues to maintain subscriber state information and route traffic. <p>When Virtual Chassis port interfaces are reconnected:</p> <ul style="list-style-type: none"> VC-M retains VC-M role. VC-L takes VC-B role. Subscribers are not affected.
Virtual Chassis port interfaces go down	Disabled	<p>When Virtual Chassis port interfaces are disconnected:</p> <ul style="list-style-type: none"> VC-M retains VC-M role, and VC-B also takes VC-M role. The result is a Virtual Chassis with two VC-M routers, each of which maintains subscriber state information. Initially, both VC-M routers have a complete list of subscribers. Because the two routers have the same configuration, the effect on subscribers, traffic patterns, behavior of external applications, and subscriber login and logout operations is unpredictable while the Virtual Chassis port interfaces are disconnected. <p>When Virtual Chassis port interfaces are reconnected:</p> <ul style="list-style-type: none"> Original VC-M before the disconnection resumes VC-M role, and original VC-B before the disconnection resumes VC-B role. Subscribers on the VC-M are preserved. Subscribers on the VC-B are purged. The subscribers preserved on the VC-M are unaffected, and all remaining subscribers are able to log back in to the router.

Table 6: Effect of Split Detection on Common Virtual Chassis Failure Scenarios (*continued*)

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

Table 6: Effect of Split Detection on Common Virtual Chassis Failure Scenarios (*continued*)

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

Related Documentation

- [Virtual Chassis Components Overview on page 5](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 13](#)
- [Mastership Election in a Virtual Chassis on page 15](#)
- [Switchover Behavior in a Virtual Chassis on page 17](#)
- [Disabling Split Detection in a Virtual Chassis Configuration on page 57](#)

CHAPTER 3

Virtual Chassis Ports and Class of Service

- [Class of Service Overview for Virtual Chassis Ports on page 25](#)
- [Guidelines for Configuring Class of Service for Virtual Chassis Ports on page 30](#)

Class of Service Overview for Virtual Chassis Ports

By default, all Virtual Chassis port interfaces in a Virtual Chassis for MX Series 3D Universal Edge Routers use a default class of service (CoS) configuration specifically tailored for Virtual Chassis ports. The default configuration, which applies to all Virtual Chassis ports in the Virtual Chassis, includes classifiers, forwarding classes, rewrite rules, and schedulers. In most cases, the default CoS configuration is adequate for your needs without requiring any additional CoS configuration.

In some cases, however, you might want to customize the traffic-control profile configuration on Virtual Chassis ports. To do so, you can configure an output traffic-control profile and apply it to all Virtual Chassis ports interfaces in the Virtual Chassis.

This topic provides an overview of the default CoS configuration for Virtual Chassis ports and helps you understand the components of the CoS configuration that you can customize.

- [Default CoS Configuration for Virtual Chassis Ports on page 25](#)
- [Supported Platforms and Maximums for CoS Configuration of Virtual Chassis Ports on page 26](#)
- [Default Classifiers for Virtual Chassis Ports on page 27](#)
- [Default Rewrite Rules for Virtual Chassis Ports on page 27](#)
- [Default Scheduler Map for Virtual Chassis Ports on page 28](#)
- [Customized CoS Configuration for Virtual Chassis Ports on page 29](#)

Default CoS Configuration for Virtual Chassis Ports

In an MX Series Virtual Chassis configuration, the Virtual Chassis ports behave like switch fabric ports to transport packets between the member routers in a Virtual Chassis. More specifically, the Virtual Chassis ports carry internal control traffic within the Virtual Chassis and forward user traffic between line cards in the router.

Like traffic on standard network port interfaces, traffic on Virtual Chassis port interfaces is mapped to one of four forwarding classes, as follows:

- Internal Virtual Chassis Control Protocol (VCCP) traffic is mapped to the network control forwarding class with the code point (IEEE 802.1p bit) value set to '111'b. You cannot change this configuration.
- Control traffic is mapped to the network control forwarding class with the code point (IEEE 802.1p bit) value set to '110'b. You cannot change this configuration.
- User traffic is mapped to the best effort, expedited forwarding, and assured forwarding traffic classes.

The CoS configuration applies globally to all Virtual Chassis ports in the Virtual Chassis. You cannot configure CoS for an individual Virtual Chassis port (such as **vcp-2/2/0**). If you create a new Virtual Chassis port, the global CoS configuration is propagated to the newly created Virtual Chassis port when the member router on which the new Virtual Chassis port resides joins the Virtual Chassis. Alternatively, you can configure CoS for the Virtual Chassis ports by configuring CoS for a standard network port, and then converting the network port to a Virtual Chassis port by issuing the **request virtual-chassis vc-port set** command.

You can convert a standard network port (for example, **xe-2/2/1**) to a Virtual Chassis port by issuing the **request virtual-chassis vc-port set** command. If the standard network port was configured with different CoS settings than the CoS configuration in effect for all Virtual Chassis ports in the Virtual Chassis, the newly converted Virtual Chassis port (**vcp-2/2/1**) uses the CoS configuration defined for all Virtual Chassis port interfaces instead of the original CoS configuration associated with the network port.

The default CoS configuration for Virtual Chassis ports provides the following benefits to keep the Virtual Chassis operating properly:

- Gives preference to internal VCCP traffic that traverses the Virtual Chassis port interfaces
- Prioritizes control traffic over user traffic on the Virtual Chassis port interfaces
- Preserves the CoS properties of each packet as it travels between member routers in the Virtual Chassis

Supported Platforms and Maximums for CoS Configuration of Virtual Chassis Ports

You can configure Virtual Chassis ports only on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces in the following MX Series 3D Universal Edge Routers with dual Routing Engines:

- MX240 3D Universal Edge Router
- MX480 3D Universal Edge Router
- MX960 3D Universal Edge Router

Trio MPC/MIC interfaces support the following maximums for forwarding classes and priority scheduling levels:

- Up to eight forwarding classes
- Up to five priority scheduling levels

Default Classifiers for Virtual Chassis Ports

Classification takes place when a packet enters a Virtual Chassis member router from a network port. For Virtual Chassis configurations that support more than two member routers, the packet is reclassified for CoS treatment according to the default IEEE 802.1p classifier rules that apply to the Virtual Chassis port as the packet travels through the intermediate member routers in the Virtual Chassis. When the packet enters the last member router in the Virtual Chassis, it is reclassified according to the original classifier rules that applied when the packet entered the Virtual Chassis from a network port.



NOTE: This reclassification behavior does not apply to an MX Series Virtual Chassis, which supports only two member routers in the current release.

Because there are no intermediate member routers between the two member routers in an MX Series Virtual Chassis, the packet is not reclassified according to the default classifier rules for the Virtual Chassis port. Instead, the original classifier rules that applied when the packet entered the Virtual Chassis on a network port are retained.

The default IEEE 802.1p classifier rules map the code point (or .1p bit) value to the forwarding class and loss priority. You can display the default IEEE 802.1p classifier rules by issuing the **show class-of-service classifier** command:

```
{master:member0-re0}

user@host> show class-of-service classifier type ieee-802.1
Classifier: ieee8021p-default, Code point type: ieee-802.1, Index: 11
  Code point      Forwarding class      Loss priority
  000             best-effort           low
  001             best-effort           high
  010             expedited-forwarding low
  011             expedited-forwarding high
  100             assured-forwarding   low
  101             assured-forwarding   high
  110             network-control    low
  111             network-control    high
```

Default Rewrite Rules for Virtual Chassis Ports

When a packet enters the Virtual Chassis from a network port, normal CoS classification takes place. If the packet exits a member router through the Virtual Chassis port to the other member router, the CoS software encapsulates the packet with a virtual LAN (VLAN) tag that contains the code point information used for CoS treatment. The code point value is assigned according to the default IEEE 802.1p rewrite rules, which map the forwarding class and loss priority value to a code point value.

You can display the default IEEE 802.1p rewrite rules by issuing the **show class-of-service rewrite-rule** command:

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

```
user@host> show class-of-service rewrite-rule type ieee-802.1
```

```
Rewrite rule: ieee8021p-default, Code point type: ieee-802.1, Index: 34
```

Forwarding class	Loss priority	Code point
best-effort	low	000
best-effort	high	001
expedited-forwarding	low	010
expedited-forwarding	high	011
assured-forwarding	low	100
assured-forwarding	high	101
network-control	low	110
network-control	high	111

Default Scheduler Map for Virtual Chassis Ports

When you create a Virtual Chassis port, it automatically functions as a hierarchical scheduler. However, you cannot explicitly configure hierarchical scheduling on Virtual Chassis ports.

Virtual Chassis ports use the same default scheduler used by standard network ports. The network control and best effort forwarding classes are both assigned low priority, and only 5 percent of the bandwidth is allocated to control traffic.

You can display the scheduler parameters and the mapping of schedulers to forwarding classes by issuing the **show class-of-service scheduler-map** command. For brevity, the following example shows only the portions of the output relevant to the default best effort (**default-be**) and default network control (**default-nc**) schedulers.

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

```
user@host> show class-of-service scheduler-map
```

```
Scheduler map: <default>, Index: 2
```

```
Scheduler: <default-be>, Forwarding class: best-effort, Index: 21
```

```
Transmit rate: 95 percent, Rate Limit: none, Buffer size: 95 percent, Buffer  
Limit: none, Priority: low
```

```
Excess Priority: low
```

```
Drop profiles:
```

Loss priority	Protocol	Index	Name
Low	any	1	<default-drop-profile>
Medium low	any	1	<default-drop-profile>
Medium high	any	1	<default-drop-profile>
High	any	1	<default-drop-profile>

```
Scheduler: <default-nc>, Forwarding class: network-control, Index: 23
```

```
Transmit rate: 5 percent, Rate Limit: none, Buffer size: 5 percent, Buffer  
Limit: none, Priority: low
```

```
Excess Priority: low
```

```
Drop profiles:
```

Loss priority	Protocol	Index	Name
Low	any	1	<default-drop-profile>
Medium low	any	1	<default-drop-profile>
Medium high	any	1	<default-drop-profile>

```

        High          any          1    <default-drop-profile>
...

```

Customized CoS Configuration for Virtual Chassis Ports

Depending on your network topology, you might want to customize the CoS configuration for Virtual Chassis ports. For example, you might want to allocate more than the default 5 percent of the Virtual Chassis port bandwidth to control traffic. Or, you might want to assign different priorities and excess rates to different forwarding classes.

Output Traffic-Control Profiles

To create a customized (nondefault) CoS configuration and apply it to all Virtual Chassis ports, you can configure an output traffic-control profile, which defines a set of traffic scheduling resources and references a scheduler map. You then apply the profile to all Virtual Chassis port interfaces. To apply the output traffic-control profile globally to all Virtual Chassis port interfaces, you must use **vcp-*** as the interface name representing all Virtual Chassis port interfaces. You cannot configure CoS for an individual Virtual Chassis port (such as **vcp-1/1/0**).

For an example that shows how to configure an output traffic-control profile customized for Virtual Chassis ports, see [“Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers” on page 99](#).

Classifiers and Rewrite Rules

Configuring nondefault IEEE 802.1p ingress classifiers and IEEE 802.1p egress rewrite rules *has no effect* in a two-member MX Series Virtual Chassis.

Because there are no intermediate routers between the two member routers in an MX Series Virtual Chassis, packets are not reclassified according to the default classifier rules for Virtual Chassis ports. Instead, the original classifier rules that applied when the packet entered the Virtual Chassis on a network port are retained, making configuration of nondefault ingress classifiers and nondefault egress rewrite rules unnecessary in the current release.

Per-Priority Shaping

Trio MPC/MIC interfaces support per-priority shaping, which enables you to configure a separate traffic shaping rate for each of the five priority scheduling levels. However, configuring per-priority shaping for Virtual Chassis ports on Trio MPC/MIC interfaces is unnecessary for the following reasons:

- The neighboring member router has exactly the same bandwidth.
- The same type of Virtual Chassis port is present at both ends of the connection.

Related Documentation

- [Guidelines for Configuring Class of Service for Virtual Chassis Ports on page 30](#)
- [Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers on page 99](#)
- [Junos OS Class of Service Configuration Guide](#)

Guidelines for Configuring Class of Service for Virtual Chassis Ports

Consider the following guidelines when you configure class of service (CoS) for Virtual Chassis ports in an MX Series Virtual Chassis:

- Virtual Chassis ports on Trio MPC/MIC interfaces support a maximum of eight forwarding classes and five priority scheduling levels.
- The same CoS configuration applies globally to all Virtual Chassis ports in the Virtual Chassis. You cannot configure CoS for an individual Virtual Chassis port (such as **vcp-3/1/0**).
- The CoS configuration is propagated to a newly created Virtual Chassis port as soon as the member router on which the new Virtual Chassis port resides joins the Virtual Chassis.
- Although Virtual Chassis ports function as hierarchical schedulers, you cannot explicitly configure hierarchical scheduling on Virtual Chassis ports.
- If you configure a nondefault output traffic-control profile to customize the CoS configuration, you must apply the profile to all Virtual Chassis port interfaces at once by using **vcp-*** as the interface name.
- Configuring nondefault IEEE 802.1p ingress classifiers and IEEE 802.1p egress rewrite rules has no effect in a two-member MX Series Virtual Chassis because the forwarding class assigned to a packet is maintained across the Virtual Chassis until the packet reaches the egress network port.
- Configuring per-priority shaping for Virtual Chassis ports is unnecessary because the neighboring member router has exactly the same bandwidth, and the same type of Virtual Chassis port is present at both ends of the connection.

Related Documentation

- [Class of Service Overview for Virtual Chassis Ports on page 25](#)
- [Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers on page 99](#)
- [Junos OS Class of Service Configuration Guide](#)

PART 2

Configuration

- [Configuration Overview on page 33](#)
- [Configuration Tasks on page 37](#)
- [Examples on page 59](#)
- [Configuration Statements on page 105](#)

CHAPTER 4

Configuration Overview

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 34](#)
- [Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 35](#)

Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis

To provide a stateful interchassis redundancy solution for MX Series routers, you can configure a Virtual Chassis. A *Virtual Chassis* interconnects two MX Series routers into a logical system that you can manage as a single network element.

To configure a Virtual Chassis for MX Series routers:

1. Prepare your site for the Virtual Chassis configuration.
[See “Preparing for a Virtual Chassis Configuration” on page 38.](#)
2. Install Junos OS licenses on the routers to be configured as members of the Virtual Chassis.
[See “Installing Junos OS Licenses on Virtual Chassis Member Routers” on page 40.](#)
3. Define configuration groups for the Virtual Chassis.
[See “Creating and Applying Configuration Groups for a Virtual Chassis” on page 42.](#)
4. Create the preprovisioned member configuration on the master router in the Virtual Chassis.
[See “Configuring Preprovisioned Member Information for a Virtual Chassis” on page 44.](#)
5. Enable graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) on both member routers.
[See “Enabling Graceful Routing Engine Switchover and Nonstop Active Routing for a Virtual Chassis” on page 46.](#)
6. Set the preprovisioned member IDs and reboot the routers in Virtual Chassis mode.

See [“Configuring Member IDs for a Virtual Chassis” on page 47](#).

7. Create the Virtual Chassis ports to interconnect the member routers, and commit the Virtual Chassis configuration on the master router.

See [“Configuring Virtual Chassis Ports to Interconnect Member Routers” on page 49](#).

8. (Optional) Verify the configuration and operation of the Virtual Chassis.

See the following topics:

- [Verifying the Status of Virtual Chassis Member Routers on page 129](#)
- [Verifying the Operation of Virtual Chassis Ports on page 129](#)
- [Verifying Neighbor Reachability for Member Routers in a Virtual Chassis on page 130](#)
- [Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis on page 130](#)
- [Viewing Information in the Virtual Chassis Control Protocol Adjacency Database on page 131](#)
- [Viewing Information in the Virtual Chassis Control Protocol Link-State Database on page 131](#)
- [Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database on page 132](#)
- [Viewing Virtual Chassis Control Protocol Routing Tables on page 132](#)
- [Viewing Virtual Chassis Control Protocol Statistics for Member Routers and Virtual Chassis Ports on page 133](#)

**Related
Documentation**

- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
- [Virtual Chassis Components Overview on page 5](#)
- [Guidelines for Configuring Virtual Chassis Ports on page 10](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers

You can delete an MX Series Virtual Chassis configuration at any time. You might want to do so if your network configuration changes, or if you want to replace one or both MX Series member routers with different MX Series routers.

To delete a Virtual Chassis configuration for MX Series routers:

1. Delete the Virtual Chassis ports from each member router.

See [“Deleting Virtual Chassis Ports in a Virtual Chassis Configuration” on page 51](#).

2. Delete the definitions and applications for the following configuration groups on each member router:

- member0-re0
 - member0-re1
 - member1-re0
 - member1-re1
3. Delete the preprovisioned member information configured at the `[edit virtual-chassis]` hierarchy level on the master router.
 4. Delete any interfaces that were configured on the member routers when the Virtual Chassis was created.
 5. Delete the Virtual Chassis member IDs to reboot each router and disable Virtual Chassis mode.
- See [“Deleting Member IDs in a Virtual Chassis Configuration” on page 52](#).



NOTE: You cannot override a Virtual Chassis configuration simply by using the `load override` command to load a different configuration on the router from an ASCII file or from terminal input, as you can with other configurations. The member ID and Virtual Chassis port definitions are not stored in the configuration file, and are still defined even after the new configuration file is loaded.

Related Documentation

- [Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 73](#)
- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
- [Virtual Chassis Components Overview on page 5](#)

Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers

You can upgrade an MX Series Virtual Chassis configuration from Junos OS Release 11.2 to a later release. This upgrade procedure assumes that both member routers in the Virtual Chassis have dual Routing Engines installed.



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

To upgrade Junos OS in a Virtual Chassis configuration consisting of two MX Series routers, each with dual Routing Engines:

1. Prepare for the upgrade.
2. Install the Junos OS software package on each of the four Routing Engines.

3. Reboot the Routing Engines to run the new Junos OS release.
4. Re-enable graceful Routing Engine switchover and nonstop active routing.

**Related
Documentation**

- [Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 85](#)
- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
- [Virtual Chassis Components Overview on page 5](#)

CHAPTER 5

Configuration Tasks

- [Preparing for a Virtual Chassis Configuration on page 38](#)
- [Installing Junos OS Licenses on Virtual Chassis Member Routers on page 40](#)
- [Creating and Applying Configuration Groups for a Virtual Chassis on page 42](#)
- [Configuring Preprovisioned Member Information for a Virtual Chassis on page 44](#)
- [Enabling Graceful Routing Engine Switchover and Nonstop Active Routing for a Virtual Chassis on page 46](#)
- [Configuring Member IDs for a Virtual Chassis on page 47](#)
- [Configuring Virtual Chassis Ports to Interconnect Member Routers on page 49](#)
- [Deleting Virtual Chassis Ports in a Virtual Chassis Configuration on page 51](#)
- [Deleting Member IDs in a Virtual Chassis Configuration on page 52](#)
- [Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 54](#)
- [Determining GRES Readiness in a Virtual Chassis Configuration on page 55](#)
- [Disabling Split Detection in a Virtual Chassis Configuration on page 57](#)

Preparing for a Virtual Chassis Configuration

Before you configure and use an MX Series Virtual Chassis, we recommend that you prepare the hardware and software in your network for the configuration.

To prepare for configuring an MX Series Virtual Chassis:

1. Make a list of the serial numbers of each router that you want to configure as part of the Virtual Chassis.

The chassis serial number is located on a label affixed to the side of the of the MX Series chassis. Alternatively, you can obtain the chassis serial number by issuing the **show chassis hardware** command, which is especially useful if you are accessing the router from a remote location. For example:

```
user@gladius> show chassis hardware
Hardware inventory:
Item           Version  Part number  Serial number  Description
Chassis                                     JN10C7135AFC  MX240
.
.
.
```

2. Note the desired function of each router in the Virtual Chassis.

In a two-router Virtual Chassis configuration, you must designate each router with the **routing-engine** role, which enables either router to function as the master or backup of the Virtual Chassis.

- The *master router* maintains the global configuration and state information for all members of the Virtual Chassis, and runs the chassis management processes.
- The *backup router* synchronizes with the master router and relays chassis control information (such as line-card presence and alarms) to the master router. If the master router is unavailable, the backup router takes mastership of the Virtual Chassis to preserve routing information and maintain network connectivity without disruption.

3. Note the member ID (0 or 1) to be assigned to each router in the Virtual Chassis.
4. Ensure that both MX Series routers in the Virtual Chassis have dual Routing Engines installed, and that all four Routing Engines in the Virtual Chassis are the same model.

For example, you cannot configure a Virtual Chassis if one member router has RE-S-2000 Routing Engines installed and the other member router has RE-S-1800 Routing Engines installed.

5. Ensure that the necessary Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces on which to configure the Virtual Chassis ports are installed and operational in each router to be configured as a member of the Virtual Chassis.



NOTE: An MX Series Virtual Chassis does not support a combination of 1-Gigabit Ethernet (ge media type) Virtual Chassis ports and 10-Gigabit Ethernet (xe media type) Virtual Chassis ports within the same Virtual Chassis. You must configure either all 10-Gigabit Ethernet Virtual Chassis ports or all 1-Gigabit Ethernet Virtual Chassis ports in the same Virtual Chassis. We recommend that you configure Virtual Chassis ports on 10-Gigabit Ethernet interfaces. This restriction has no effect on access ports or uplink ports in an MX Series Virtual Chassis configuration.

6. If MX Series Enhanced Queuing IP Services DPCs (DPCE-R-Q model numbers) or MX Series Enhanced Queuing Ethernet Services DPCs (DPCE-X-Q model numbers) are installed in a router to be configured as a member of the Virtual Chassis, make sure these DPCs are offline before you configure the Virtual Chassis. Otherwise, the MX Series Virtual Chassis configuration will not function.



NOTE: MX Series Enhanced Queuing IP Services DPCs (DPCE-R-Q model numbers) and MX Series Enhanced Queuing Ethernet Services DPCs (DPCE-X-Q model numbers) do not interoperate with features of the MX Series Virtual Chassis.

7. Determine the desired location of the dedicated Virtual Chassis ports on both member routers, and use the Virtual Chassis ports to physically interconnect the member routers in a point-to-point topology.
8. Ensure that both MX Series routers to be configured as a member of the Virtual Chassis are running the same Junos OS release, and have basic network connectivity.

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Guidelines for Configuring Virtual Chassis Ports on page 10](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Installing Junos OS Licenses on Virtual Chassis Member Routers

To enable some Junos OS features or router scaling levels, you might have to purchase, install, and manage separate software license packs. The presence on the router of the appropriate software license keys (passwords) determines whether you can configure and use certain features or configure a feature to a predetermined scale.

Before you configure an MX Series Virtual Chassis, install the following Junos OS software licenses on each MX Series router to be configured as a member of the Virtual Chassis:

- **MX Virtual Chassis Redundancy Feature Pack**—You must purchase and install a unique MX Virtual Chassis Redundancy Feature Pack for each member router in the Virtual Chassis. If you issue the **request virtual-chassis member-id set**, **request virtual-chassis member-id delete**, **request virtual-chassis vc-port set**, or **request virtual-chassis vc-port delete** command to set or delete member IDs or Virtual Chassis ports without first installing an MX Virtual Chassis Redundancy Feature Pack on both member routers, the software displays a warning message that you are operating without a valid Virtual Chassis software license.
- **Junos OS feature licenses**—Purchase and install the appropriate Junos OS feature licenses to enable use of a particular software feature or scaling level in your network. You must install the required feature licenses on each member router in the Virtual Chassis.

Before you begin:

- Prepare your site for the Virtual Chassis configuration.
See [“Preparing for a Virtual Chassis Configuration” on page 38](#).
- Familiarize yourself with the procedures for installing and managing Junos OS licenses.
See [Installation and Upgrade Guide](#).

To install Junos OS licenses on each member router in the Virtual Chassis:

1. Install the required licenses on the MX Series router to be designated as the protocol master for the Virtual Chassis.
 - a. Install the MX Virtual Chassis Redundancy Feature Pack.
 - b. Install the Junos OS feature licenses required for your software feature or scaling level.
2. Install the required licenses on the MX Series router to be designated as the protocol backup for the Virtual Chassis.
 - a. Install the MX Virtual Chassis Redundancy Feature Pack.
 - b. Install the Junos OS feature licenses required for your software feature or scaling level.
3. (Optional) Verify the license installation on each member router.

For example:

```
user@host> show system license
```

License usage:

Feature name	Licenses used	Licenses installed	Licenses needed	Expiry
subscriber-accounting	0	1	0	permanent
subscriber-authentication	0	1	0	permanent
subscriber-address-assignment	0	1	0	permanent
subscriber-vlan	0	1	0	permanent
subscriber-ip	0	1	0	permanent
scale-subscriber	0	256000	0	permanent
scale-l2tp	0	1000	0	permanent
scale-mobile-ip	0	1000	0	permanent
virtual-chassis	0	1	0	permanent

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)
- Software Features That Require Licenses on MX Series Routers Only
- Installation and Upgrade Guide

Creating and Applying Configuration Groups for a Virtual Chassis

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

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



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

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

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

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

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

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

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

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

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

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

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

```
user@gladius# delete groups re1
```

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

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

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

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

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

7. Commit the configuration.



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

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

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)
- For more information about creating and managing configuration groups, see the CLI User Guide

Configuring Preprovisioned Member Information for a Virtual Chassis

To configure a Virtual Chassis for MX Series routers, you must create a preprovisioned configuration on the master router by including the **virtual-chassis** stanza at the **[edit virtual-chassis]** hierarchy level. The preprovisioned configuration specifies the chassis serial number, member ID, and role for both member routers in the Virtual Chassis.

When a new member router joins the Virtual Chassis, the software compares its serial number against the values specified in the preprovisioned configuration. If the serial number of a joining router does not match any of the configured serial numbers, the software prevents that router from becoming a member of the Virtual Chassis.

To configure the preprovisioned member information for an MX Series Virtual Chassis:

1. Specify that you want to create a preprovisioned Virtual Chassis configuration.

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

2. Configure the member ID (0 or 1), role (**routing-engine**), and chassis serial number for each member router in the Virtual Chassis.

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



NOTE: In a two-member MX Series Virtual Chassis configuration, you must assign the **routing-engine** role to each router. The **routing-engine** role enables the router to function either as the master router or backup router of the Virtual Chassis.

3. Disable detection of a split in the Virtual Chassis configuration. (By default, split detection in an MX Series Virtual Chassis is enabled.)

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



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

4. (Optional) Enable tracing of Virtual Chassis operations.

For example:

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

```

user@gladius# set traceoptions file filename
user@gladius# set traceoptions file size maximum-file-size
user@gladius# set traceoptions flag flag

```

5. Commit the configuration.



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

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

The following example shows an MX Series Virtual Chassis preprovisioned configuration for two member routers.

```

[edit virtual-chassis]
user@gladius# show
preprovisioned;
no-split-detection;
traceoptions {
    file vccp size 10m;
    flag all;
}
member 0 {
    role routing-engine;
    serial-number JN115FDADAFB;
}
member 1 {
    role routing-engine;
    serial-number JN10C78D1AFC;
}

```

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Enabling Graceful Routing Engine Switchover and Nonstop Active Routing for a Virtual Chassis

Before you configure member IDs and Virtual Chassis ports, you must enable graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) on both member routers in the Virtual Chassis.

To enable graceful Routing Engine switchover and nonstop active routing:

1. Enable graceful Routing Engine switchover and nonstop active routing on member 0 (**gladius**):
 - a. Log in to the console on member 0.
 - b. Enable graceful switchover.

```
[edit chassis redundancy]
user@gladius# set graceful-switchover
```
 - c. Enable nonstop active routing.

```
[edit routing-options]
user@gladius# set nonstop-routing
```
 - d. Commit the configuration on member 0.

```
[edit system]
user@gladius# commit synchronize
```
2. Enable graceful Routing Engine switchover and nonstop active routing on member 1 (**trefoil**):
 - a. Log in to the console on member 1.
 - b. Enable graceful switchover.

```
[edit chassis redundancy]
user@trefoil# set graceful-switchover
```
 - c. Enable nonstop active routing.

```
[edit routing-options]
user@trefoil# set nonstop-routing
```
 - d. Commit the configuration on member 1.

```
[edit system]
user@trefoil# commit synchronize
```



NOTE: When you configure nonstop active routing, you must include the `commit synchronize` statement at the `[edit system]` hierarchy level. Otherwise, the commit operation fails.

For an MX Series Virtual Chassis, the `force` option is the default and only behavior when you use the `commit synchronize` statement. Including the `commit synchronize` statement for an MX Series Virtual Chassis configuration has the same effect as including the `commit synchronize force` statement.

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)
- [Configuring Graceful Routing Engine Switchover](#)
- [Configuring Nonstop Active Routing](#)

Configuring Member IDs for a Virtual Chassis

After you commit the preprovisioned configuration on the master router, you must assign the preprovisioned member IDs to both MX Series routers in the Virtual Chassis by using the `request virtual-chassis member-id set` command. Assigning the member ID causes the router to reboot in preparation for forming the Virtual Chassis.



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

To configure the member ID and reboot each MX Series router in Virtual Chassis mode:

1. Set the member ID on the router configured as **member 0**.

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

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

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

2. Set the member ID on the router configured as **member 1**.

```
user@hostB> request virtual-chassis member-id set member 1
```

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

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

3. (Optional) Verify the member ID configuration for **member 0**.

For example:

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

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

```
Preprovisioned Virtual Chassis
```

```
Virtual Chassis ID: 4f2b.1aa0.de08
```

				Mastership		Neighbor
List						
Member ID	Status	Serial No	Model	priority	Role	ID
Interface						
0 (FPC 0- 11)	Prsnt	JN10C7135AFC	mx240	129	Master*	

4. (Optional) Verify the member ID configuration for **member 1**.

For example:

```
Amnesiac (ttyd0)
```

```
login: user
```

```
Password:
```

```
...
```

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

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

```
Virtual Chassis ID: ef98.2c6c.f7f7
```

				Mastership		Neighbor
List						
Member ID	Status	Serial No	Model	priority	Role	ID
Interface						
1 (FPC 12- 23)	Prsnt	JN115D117AFB	mx480	128	Master*	



NOTE: At this point in the configuration procedure, all line cards are offline, and the routers are each designated with the Master role because they are not yet interconnected as a fully formed Virtual Chassis. In addition, member 1 remains in Amnesiac state (has no defined configuration) until the Virtual Chassis forms and the configuration is committed.

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Configuring Virtual Chassis Ports to Interconnect Member Routers

To interconnect the member routers in an MX Series Virtual Chassis, you must use the **request virtual-chassis vc-port set** command to configure Virtual Chassis ports on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces. After it is configured, a Virtual Chassis port is dedicated to the task of interconnecting member routers, and is no longer available for configuration as a standard network port.



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

To configure Virtual Chassis ports on Trio MPC/MIC interfaces to interconnect the member routers in an MX Series Virtual Chassis:

1. Configure the Virtual Chassis ports on the router configured as member 0.

- a. Configure the first Virtual Chassis port that connects to member 1.

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

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

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

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

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

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

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

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

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

2. Configure the Virtual Chassis ports on the router configured as member 1.

- a. Configure the first Virtual Chassis port that connects to member 0.

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

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

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

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

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

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



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

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

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

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

```
user@hostA> show virtual-chassis status  
{master:member0-re0}
```

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

4. Commit the configuration on the master router.

```
{master:member0-re0}[edit system]  
user@hostA# commit synchronize
```

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

Related Documentation

- [Guidelines for Configuring Virtual Chassis Ports on page 10](#)
- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Deleting Virtual Chassis Ports in a Virtual Chassis Configuration

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



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

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

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

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

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

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

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

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

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

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

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

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

When you delete a Virtual Chassis port, its name (**vcp-slot/pic/port**) no longer appears in the output of the **show virtual-chassis vc-port** command. For example, the following output for the **show virtual-chassis vc-port** command on each member router confirms that all Virtual Chassis ports have been deleted from both member routers.

For member 0 (**host1**):

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

```
user@host1> show virtual-chassis vc-port all-members  
member0:
```

```
-----
```

For member 1 (**host2**):

```
{backup:member1-re0}

user@host2> show virtual-chassis vc-port all-members
member1:
-----
```



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

Related Documentation

- [Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 34](#)
- [Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 73](#)
- [Guidelines for Configuring Virtual Chassis Ports on page 10](#)

Deleting Member IDs in a Virtual Chassis Configuration

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



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

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

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

```
{master:member0-re0}

user@host1> request virtual-chassis member-id delete
This command will disable virtual-chassis mode and reboot the system.
Continue? [yes,no] (no) yes

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

{master:member0-re0}
user@host1>

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

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

```
{master:member1-re0}
user@host2> request virtual-chassis member-id delete
This command will disable virtual-chassis mode and reboot the system.
Continue? [yes,no] (no) yes

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

{master:member1-re0}
user@host2>

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

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

For example:

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

**Related
Documentation**

- [Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 34](#)
- [Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 73](#)

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

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

Before you begin:

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

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

Error: mastership switch request NOT honored, backup not ready

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

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

To switch the global master and backup roles in an MX Series Virtual Chassis:

- Issue the **request virtual-chassis routing-engine master switch** command from the Virtual Chassis master router:

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

```
user@host1> request virtual-chassis routing-engine master switch
Do you want to continue ? [yes,no] (no) yes
```

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

error: Virtual Chassis member is not the protocol master

Issuing the **request virtual-chassis routing-engine master switch** command from the Virtual Chassis master router causes the global role transitions listed in [Table 7 on page 54](#).

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

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

Table 7: Virtual Chassis Global Role Transitions Before and After Mastership Switchover (*continued*)

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

The local roles (**master** and **standby**, or **m** and **s**) of the Routing Engines do not change after you issue the **request virtual-chassis routing-engine master switch** command. For example, as shown in [Table 7 on page 54](#), the master Routing Engine in the Virtual Chassis master router (VC-Mm) remains the master Routing Engine in the Virtual Chassis backup router (VC-Bm) after the switchover.

Related Documentation

- [Virtual Chassis Components Overview on page 5](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 13](#)
- [Mastership Election in a Virtual Chassis on page 15](#)
- [Switchover Behavior in a Virtual Chassis on page 17](#)

Determining GRES Readiness in a Virtual Chassis Configuration

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

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

To determine whether the member routers in an MX Series Virtual Chassis are ready for GRES from a database synchronization perspective:

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

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

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

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

2. Review the results of the **request virtual-chassis routing-engine master switch check** command to determine whether the member routers in the MX Series Virtual Chassis are ready for a GRES operation from a database synchronization perspective.

- If the member routers are ready for GRES, the **request virtual-chassis routing-engine master switch check** command returns the command prompt and displays no output.

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

```
user@host> request virtual-chassis routing-engine master switch check  
{master:member0-re0}
```

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

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

```
user@host> request virtual-chassis routing-engine master switch check  
error: chassisd Not ready for mastership switch, try after 217 secs.  
mastership switch request NOT honored, backup not ready
```

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

Related Documentation

- [Switchover Behavior in a Virtual Chassis on page 17](#)
- [Virtual Chassis Components Overview on page 5](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 13](#)
- [Understanding Graceful Routing Engine Switchover in the Junos OS](#)

Disabling Split Detection in a Virtual Chassis Configuration

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

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

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



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

To disable split detection in an MX Series Virtual Chassis:

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

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

2. Commit the configuration.

Disabling split detection causes different results for different types of Virtual Chassis failures. For information, see [“Split Detection Behavior in a Virtual Chassis” on page 19](#).

Related Documentation

- [Split Detection Behavior in a Virtual Chassis on page 19](#)
- [Global Roles and Local Roles in a Virtual Chassis on page 13](#)
- [Switchover Behavior in a Virtual Chassis on page 17](#)
- [Virtual Chassis Components Overview on page 5](#)

CHAPTER 6

Examples

- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)
- [Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 73](#)
- [Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 85](#)
- [Example: Replacing a Routing Engine in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 90](#)
- [Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers on page 99](#)

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

To provide interchassis redundancy for MX Series 3D Universal Edge Routers, you can configure a Virtual Chassis. A *Virtual Chassis* configuration interconnects two MX Series routers into a logical system that you can manage as a single network element. The member routers in a Virtual Chassis are interconnected by means of Virtual Chassis ports that you configure on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces (network ports) on each MX Series router.

This example describes how to set up and configure a Virtual Chassis consisting of two MX Series routers:

- [Requirements on page 59](#)
- [Overview and Topology on page 60](#)
- [Configuration on page 62](#)
- [Verification on page 71](#)

Requirements

This example uses the following software and hardware components:

- Junos OS Release 11.2
- One MX240 3D Universal Edge Router

- One MX480 3D Universal Edge Router

See [Table 8 on page 61](#) for information about the hardware installed in each MX Series router.



NOTE: MX Series Enhanced Queuing IP Services DPCs (DPCE-R-Q model numbers) and MX Series Enhanced Queuing Ethernet Services DPCs (DPCE-X-Q model numbers) do not interoperate with features of the MX Series Virtual Chassis. If any MX Series Enhanced Queuing DPCs are installed in a router to be configured as a member of a Virtual Chassis, you must ensure that these DPCs are offline before you configure the Virtual Chassis.

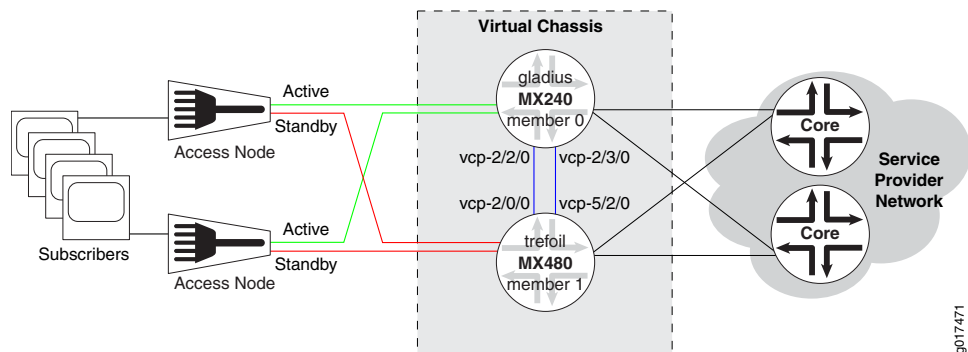
Overview and Topology

To configure the Virtual Chassis shown in this example, you must create a preprovisioned configuration at the **[edit virtual-chassis]** hierarchy level on the router to be designated as the master of the Virtual Chassis. The preprovisioned configuration includes the serial number, member ID, and role for each member router (also known as member chassis) in the Virtual Chassis. When a new member router joins the Virtual Chassis, the software compares its serial number against the values specified in the preprovisioned configuration. If the serial number of a joining router does not match any of the configured serial numbers, the software prevents that router from becoming a member of the Virtual Chassis.

After you commit the preprovisioned configuration on the master router, you must assign the preprovisioned member IDs by issuing the **request virtual-chassis member-id set** administrative command on each router, which causes the router to reboot. When the reboot is complete, you create one or more Virtual Chassis ports by issuing the **request virtual-chassis vc-port set** administrative command on each router. The Virtual Chassis forms when the line cards in both member routers are back online.

This example configures a Virtual Chassis that interconnects two MX Series routers, and uses the basic topology shown in [Figure 2 on page 60](#). For redundancy, two Virtual Chassis ports are configured on each member router.

Figure 2: Sample Topology for a Virtual Chassis with Two MX Series Routers



g017471

Table 8 on page 61 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis. You use some of these settings in the preprovisioned configuration and when you assign the member IDs and create the Virtual Chassis ports.

Table 8: Components of the Sample MX Series Virtual Chassis

Router Name	Hardware	Serial Number	Member ID	Role	Virtual Chassis Ports	Network Port Slot Numbering
gladius	MX240 router with: <ul style="list-style-type: none"> • 60-Gigabit Ethernet Enhanced Queuing MPC • 20-port Gigabit Ethernet MIC with SFP • 4-port 10-Gigabit Ethernet MIC with XFP • Master RE-S-2000 Routing Engine in slot 0 (represented in example as member0-re0) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as member0-re1) 	JN10C7135AFC	0	routing-engine (master)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11
trefoil	MX480 router with: <ul style="list-style-type: none"> • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Master RE-S-2000 Routing Engine in slot 0 (represented in example as member1-re0) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as member1-re1) 	JN115D117AFB	1	routing-engine (backup)	vcp-2/0/0 vcp-5/2/0	FPC 12 – 23 (offset = 12)

Configuration

To configure a Virtual Chassis consisting of two MX Series routers, perform these tasks:

- [Preparing for the Virtual Chassis Configuration on page 62](#)
- [Creating and Applying Configuration Groups for the Virtual Chassis on page 64](#)
- [Configuring Preprovisioned Member Information for the Virtual Chassis on page 66](#)
- [Enabling Graceful Routing Engine Switchover and Nonstop Active Routing on page 67](#)
- [Configuring Member IDs and Rebooting the Routers to Enable Virtual Chassis Mode on page 68](#)
- [Configuring Virtual Chassis Ports to Interconnect Member Routers on page 70](#)

Preparing for the Virtual Chassis Configuration

Step-by-Step Procedure

To prepare for configuring an MX Series Virtual Chassis:

1. Make a list of the serial numbers of both routers that you want to configure as part of the Virtual Chassis.

The chassis serial number is located on a label affixed to the side of the of the MX Series chassis. Alternatively, you can obtain the chassis serial number by issuing the **show chassis hardware** command, which is especially useful if you are accessing the router from a remote location. For example:

```
user@gladius> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number  Description
Chassis
.
.
.
Fan Tray 0       REV 01   710-021113   JT0119         MX240 Fan Tray
```

2. Note the desired role (**routing-engine**) for each router in the Virtual Chassis.

In a two-router Virtual Chassis configuration, you must designate each router with the **routing-engine** role, which enables either router to function as the master or backup of the Virtual Chassis.

- The *master router* maintains the global configuration and state information for all members of the Virtual Chassis, and runs the chassis management processes.
- The *backup router* synchronizes with the master router and relays chassis control information (such as line-card presence and alarms) to the master router. If the master router is unavailable, the backup router takes mastership of the Virtual Chassis to preserve routing information and maintain network connectivity without disruption.

3. Note the member ID (0 or 1) to be assigned to each router in the Virtual Chassis.

In this example, the master router is assigned member ID 0, and the backup router is assigned member ID 1.

4. Ensure that both MX Series routers in the Virtual Chassis have dual Routing Engines installed, and that all four Routing Engines in the Virtual Chassis are the same model.

For example, you cannot configure a Virtual Chassis if one member router has RE-S-2000 Routing Engines installed and the other member router has RE-S-1800 Routing Engines installed.

5. Ensure that the necessary Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces on which to configure the Virtual Chassis ports are installed and operational in each router to be configured as a member of the Virtual Chassis.



NOTE: An MX Series Virtual Chassis does not support a combination of 1-Gigabit Ethernet (ge media type) Virtual Chassis ports and 10-Gigabit Ethernet (xe media type) Virtual Chassis ports within the same Virtual Chassis. You must configure either all 10-Gigabit Ethernet Virtual Chassis ports or all 1-Gigabit Ethernet Virtual Chassis ports in the same Virtual Chassis. We recommend that you configure Virtual Chassis ports on 10-Gigabit Ethernet interfaces. This restriction has no effect on access ports or uplink ports in an MX Series Virtual Chassis configuration.

6. If MX Series Enhanced Queuing IP Services DPCs (DPCE-R-Q model numbers) or MX Series Enhanced Queuing Ethernet Services DPCs (DPCE-X-Q model numbers) are installed in a router to be configured as a member of the Virtual Chassis, make sure these DPCs are offline before you configure the Virtual Chassis. Otherwise, the MX Series Virtual Chassis configuration will not function.



NOTE: MX Series Enhanced Queuing IP Services DPCs (DPCE-R-Q model numbers) and MX Series Enhanced Queuing Ethernet Services DPCs (DPCE-X-Q model numbers) do not interoperate with features of the MX Series Virtual Chassis.

7. Determine the desired location of the dedicated Virtual Chassis ports on both member routers, and use the Virtual Chassis ports to physically interconnect the member routers in a point-to-point topology.
8. Ensure that both MX Series routers to be configured as members of the Virtual Chassis are running the same Junos OS release, and have basic network connectivity.
9. Install the MX Virtual Chassis Redundancy Feature Pack license on each router to be configured as part of the Virtual Chassis.
10. Install the necessary Junos OS feature licenses on each router to be configured as part of the Virtual Chassis.

Creating and Applying Configuration Groups for the Virtual Chassis

Step-by-Step Procedure For a Virtual Chassis configuration consisting of two MX Series routers, each of which supports dual Routing Engines, you must create and apply the following configuration groups on the router to be designated as the master of the Virtual Chassis instead of using the standard `re0` and `re1` configuration groups:

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



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

To create and apply configuration group information for the Virtual Chassis:

1. Log in to the console on member 0 (`gladius`).
2. In the console window on member 0, create and apply the `member0-re0` configuration group.

```
[edit]
user@gladius# copy groups re0 to member0-re0
user@gladius# set apply-groups member0-re0
```
3. Delete the standard `re0` configuration group from the global configuration on member 0.

```
[edit]
user@gladius# delete apply-groups re0
user@gladius# delete groups re0
```
4. Create and apply the `member0-re1` configuration group on member 0.

```
[edit]
user@gladius# copy groups re1 to member0-re1
user@gladius# set apply-groups member0-re1
```
5. Delete the standard `re1` configuration group from the global configuration on member 0.

```
[edit]
user@gladius# delete apply-groups re1
user@gladius# delete groups re1
```
6. Create and apply the `member1-re0` configuration information on member 0.

```
[edit]
user@gladius# set groups member1-re0 system host-name trefoil
user@gladius# set groups member1-re0 system backup-router 10.9.0.1
```



```

user@gladius# set groups member1-re0 system backup-router destination
172.16.0.0/12
user@gladius# set groups member1-re0 system backup-router destination
10.9.0.0/16
...
user@gladius# set groups member1-re0 interfaces fxp0 unit 0 family inet address
10.9.3.97/21
user@gladius# set apply-groups member1-re0

```

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

7. Create and apply the **member1-re1** configuration information on member 0.

```

[edit]
user@gladius# set groups member1-re1 system host-name trefoil
user@gladius# set groups member1-re1 system backup-router 10.9.0.1
user@gladius# set groups member1-re1 system backup-router destination
172.16.0.0/12
user@gladius# set groups member1-re1 system backup-router destination 10.9.0.0/16
...
user@gladius# set groups member1-re1 interfaces fxp0 unit 0 family inet address
10.9.3.98/21
user@gladius# set apply-groups member1-re1

```

8. Commit the configuration on member 0.



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

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

Results Display the results of the configuration.

```

[edit]
user@gladius# show groups ?
Possible completions:
<[Enter]>      Execute this command
<group_name>  Group name
global        Group name
member0-re0   Group name
member0-re1   Group name
member1-re0   Group name
member1-re1   Group name
|             Pipe through a command

[edit]
user@gladius# show apply-groups

```

```
apply-groups [ global member0-re0 member0-re1 member1-re0 member1-re1 ];
```

Configuring Preprovisioned Member Information for the Virtual Chassis

Step-by-Step Procedure

To configure the preprovisioned member information on member 0 (**gladius**):

1. Log in to the console on member 0.
2. Specify that you want to create a preprovisioned Virtual Chassis configuration.

```
[edit virtual-chassis]  
user@gladius# set preprovisioned
```
3. Configure the member ID (0 or 1), role (**routing-engine**), and chassis serial number for each member router in the Virtual Chassis.

```
[edit virtual-chassis]  
user@gladius# set member 0 role routing-engine serial-number JN10C7135AFC  
user@gladius# set member 1 role routing-engine serial-number JN115D117AFB
```
4. Disable detection of a split in the Virtual Chassis configuration. (By default, split detection in an MX Series Virtual Chassis is enabled.)

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



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

5. (Optional) Enable tracing of Virtual Chassis operations.

```
[edit virtual-chassis]  
user@gladius# set traceoptions file vccp  
user@gladius# set traceoptions file size 100m  
user@gladius# set traceoptions flag all
```
6. Commit the configuration.

Results Display the results of the configuration.

```
[edit virtual-chassis]  
user@gladius# show  
preprovisioned;  
no-split-detection;  
traceoptions {  
  file vccp size 100m;  
  flag all;  
}  
member 0 {  
  role routing-engine;
```

```

    serial-number JN10C7135AFC;
  }
  member 1 {
    role routing-engine;
    serial-number JN115D117AFB;
  }

```

Enabling Graceful Routing Engine Switchover and Nonstop Active Routing

Step-by-Step Procedure Before you configure member IDs and Virtual Chassis ports, you must enable graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) on both member routers in the Virtual Chassis.

To enable graceful Routing Engine switchover and nonstop active routing:

1. Enable graceful Routing Engine switchover and nonstop active routing on member 0 (**gladius**):
 - a. Log in to the console on member 0.
 - b. Enable graceful switchover.


```

[edit chassis redundancy]
user@gladius# set graceful-switchover
          
```
 - c. Enable nonstop active routing.


```

[edit routing-options]
user@gladius# set nonstop-routing
          
```
 - d. Commit the configuration on member 0.


```

[edit system]
user@gladius# commit synchronize
          
```
2. Enable graceful Routing Engine switchover and nonstop active routing on member 1 (**trefoil**):
 - a. Log in to the console on member 1.
 - b. Enable graceful switchover.


```

[edit chassis redundancy]
user@trefoil# set graceful-switchover
          
```
 - c. Enable nonstop active routing.


```

[edit routing-options]
user@trefoil# set nonstop-routing
          
```
 - d. Commit the configuration on member 1.


```

[edit system]
user@trefoil# commit synchronize
          
```



NOTE: When you configure nonstop active routing, you must include the `commit synchronize` statement at the `[edit system]` hierarchy level. Otherwise, the commit operation fails.

For an MX Series Virtual Chassis, the `force` option is the default and only behavior when you use the `commit synchronize` statement. Including the `commit synchronize` statement for an MX Series Virtual Chassis configuration has the same effect as including the `commit synchronize force` statement.

Configuring Member IDs and Rebooting the Routers to Enable Virtual Chassis Mode

Step-by-Step Procedure

To configure (set) the preprovisioned member ID for each MX Series router in the Virtual Chassis, use the `request virtual-chassis member-id set` command. Assigning the member ID causes the router to reboot in preparation for forming the Virtual Chassis.



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

To configure the member ID and reboot each router to enable Virtual Chassis mode:

1. Log in to the console on member 0 (**gladius**).
2. Set the member ID on member 0.

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

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

Continue? [yes,no] yes

Issuing the `request virtual-chassis member-id` command causes the router to reboot in preparation for membership in the Virtual Chassis.

3. Log in to the console on member 1 (**trefoil**).
4. Set the member ID on member 1.

```
user@trefoil> request virtual-chassis member-id set member 1
```

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

Continue? [yes,no] yes

Results Display the results of the configuration on each router. At this point in the procedure, all line cards are offline, and the routers are each designated with the **Master** role because they are not yet interconnected as a fully formed Virtual Chassis. In addition, member 1

(**trefoil**) remains in Amnesiac state (has no defined configuration) until the Virtual Chassis forms and the configuration is committed.

For member 0 (**gladius**):

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

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

```
Preprovisioned Virtual Chassis
```

```
Virtual Chassis ID: 4f2b.1aa0.de08
```

				Mastership		Neighbor List	
Member ID	Status	Serial No	Model	priority	Role	ID	Interface
0 (FPC 0- 11)	Prsnt	JN10C7135AFC	mx240	129	Master*		

For member 1 (**trefoil**):

```
Amnesiac (ttyd0)
```

```
login: user
```

```
Password:
```

```
...
```

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

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

```
Virtual Chassis ID: eabf.4e50.91e6
```

```
Virtual Chassis Mode: Disabled
```

				Mastership		Neighbor List	
Member ID	Status	Serial No	Model	priority	Role	ID	Interface
1 (FPC 12- 23)	Prsnt	JN115D117AFB	mx480	128	Master*		

Configuring Virtual Chassis Ports to Interconnect Member Routers

Step-by-Step Procedure To interconnect the member routers in an MX Series Virtual Chassis, use the **request virtual-chassis vc-port set** command to configure (set) Virtual Chassis ports on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces.



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

To configure Virtual Chassis ports on Trio MPC/MIC interfaces to connect the member routers in the Virtual Chassis:

1. Configure the Virtual Chassis ports on member 0 (**gladius**).
 - a. Log in to the console on member 0.
 - b. Configure the first Virtual Chassis port that connects to member 1 (**trefoil**).

```
{master:member0-re0}

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

After the Virtual Chassis port is created, it is renamed **vcp-slot/pic/port** (for example, **vcp-2/2/0**), and the line card associated with that port comes online. The line cards in the other member router remain offline until the Virtual Chassis forms. Each Virtual Chassis port is dedicated to the task of interconnecting member routers in a Virtual Chassis, and is no longer available for configuration as a standard network port.
 - c. When **vcp-2/2/0** is up, configure the second Virtual Chassis port that connects to member 1.

```
{master:member0-re0}

user@gladius> request virtual-chassis vc-port set fpc-slot 2 pic-slot 3 port 0
vc-port successfully set
```
2. Configure the Virtual Chassis ports on member 1 (**trefoil**).
 - a. Log in to the console on member 1.
 - b. Configure the first Virtual Chassis port that connects to member 0 (**gladius**).

```
{master:member1-re0}

user@trefoil> request virtual-chassis vc-port set fpc-slot 2 pic-slot 0 port 0
vc-port successfully set
```
 - c. When **vcp-2/0/0** is up, configure the second Virtual Chassis port that connects to member 0.

```
{master:member1-re0}

user@trefoil> request virtual-chassis vc-port set fpc-slot 5 pic-slot 2 port 0
```

```
vc-port successfully set
```

When all of the line cards in all of the member routers are online, and the Virtual Chassis has formed, you can issue Virtual Chassis commands from the terminal window of the master router (**gladius**).

3. Verify that the Virtual Chassis is properly configured and operational.

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

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

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

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

See the Verification section for information about interpreting the output of these commands.

4. Commit the configuration on the master router.

```
{master:member0-re0}[edit system]
```

```
user@gladius# commit synchronize
```

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

Verification

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

- [Verifying the Member IDs and Roles of the Virtual Chassis Members on page 71](#)
- [Verifying the Operation of the Virtual Chassis Ports on page 72](#)
- [Verifying Neighbor Reachability on page 72](#)

Verifying the Member IDs and Roles of the Virtual Chassis Members

Purpose Verify that the member IDs and roles of the routers belonging to the Virtual Chassis are properly configured.

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

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

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

```
Preprovisioned Virtual Chassis
```

```
Virtual Chassis ID: a5b6.be0c.9525
```

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID Interface
0 (FPC 0- 11)	Prsnt	JN10C7135AFC	mx240	129	Master*	1 vcp-2/2/0 1 vcp-2/3/0
1 (FPC 12- 23)	Prsnt	JN115D117AFB	mx480	129	Backup	0 vcp-2/0/0 0 vcp-5/2/0

Meaning The value **Prsnt** in the **Status** column of the output confirms that the member routers specified in the preprovisioned configuration are currently connected to the Virtual Chassis. The display shows that member 0 (**gladius**) and member 1 (**trefoil**), which were both

configured with the **routing-engine** role, are functioning as the master router and backup router of the Virtual Chassis, respectively. The **Neighbor List** displays the interconnections between the member routers by means of the Virtual Chassis ports. For example, member 0 is connected to member 1 through **vcp-2/2/0** and **vcp-2/3/0**. The asterisk (*) following **Master** denotes the router on which the command was issued. The **Mastership priority** value is assigned by the software and is not configurable in the current release.

Verifying the Operation of the Virtual Chassis Ports

Purpose Verify that the Virtual Chassis ports are properly configured and operational.

Action Display the status of the Virtual Chassis ports for both members of the Virtual Chassis.

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

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

```
member0:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
2/2/0	Configured	3	Up	10000	1	vcp-2/0/0
2/3/0	Configured	3	Up	10000	1	vcp-5/2/0

```
member1:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID	Interface
2/0/0	Configured	3	Up	10000	0	vcp-2/2/0
5/2/0	Configured	3	Up	10000	0	vcp-2/3/0

Meaning The output confirms that the Virtual Chassis ports you configured are operational. For each member router, the **Interface or Slot/PIC/Port** column shows the location of the Virtual Chassis ports configured on that router. For example, the Virtual Chassis ports on **member0-re0 (gladius)** are **vcp-2/2/0** and **vcp-2/3/0**. In the **Trunk ID** column, the value **3** indicates that a trunk has formed; if a trunk is not present, this field displays the value **-1**. In the **Status** column, the value **Up** confirms that the interfaces associated with the Virtual Chassis ports are operational. The **Speed** column displays the speed of the Virtual Chassis port interface. The **Neighbor ID/Interface** column displays the member IDs and Virtual Chassis port interfaces that connect to this router. For example, the connections to member 0 (**gladius**) are through **vcp-2/0/0** and **vcp-5/2/0** on member 1 (**trefoil**).

Verifying Neighbor Reachability

Purpose Verify that each member router in the Virtual Chassis can reach the neighbor routers to which it is connected.

Action Display the neighbor reachability information for both member routers in the Virtual Chassis.

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

```
user@gladius> show virtual-chassis active-topology all-members
member0:
```

Destination ID	Next-hop
1	1(vcp-2/2/0.32768)

```
member1:
```

Destination ID	Next-hop
0	0(vcp-2/0/0.32768)

Meaning The output confirms that each member router in the Virtual Chassis has a path to reach the neighbors to which it is connected. For each member router, the **Destination ID** specifies the member ID of the destination (neighbor) router. The **Next-hop** column displays the member ID and Virtual Chassis port interface of the next-hop to which packets for the destination ID are forwarded. For example, the next-hop from member 0 (**gladius**) to member 1 (**trefoil**) is through Virtual Chassis port interface **vcp-2/2/0.32768**.

- Related Documentation**
- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
 - [Virtual Chassis Components Overview on page 5](#)
 - [Guidelines for Configuring Virtual Chassis Ports on page 10](#)
 - [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)

Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers

You can delete an MX Series Virtual Chassis configuration at any time. You might want to do so if your network configuration changes, or if you want to replace one or both MX Series member routers in the Virtual Chassis with different MX Series routers. After you delete the Virtual Chassis configuration, the routers that were formerly members of the Virtual Chassis function as two independent routers.

This example describes how to delete a Virtual Chassis configuration consisting of two MX Series routers:

- [Requirements on page 74](#)
- [Overview and Topology on page 74](#)
- [Configuration on page 76](#)
- [Verification on page 83](#)

Requirements

This example uses the following software and hardware components:

- Junos OS Release 11.2
- One MX240 3D Universal Edge Router with dual Routing Engines
- One MX480 3D Universal Edge Router with dual Routing Engines

See [Table 9 on page 75](#) for information about the hardware installed in each MX Series router.

Overview and Topology

To delete an MX Series Virtual Chassis configuration, you must:

1. Delete all Virtual Chassis ports.
2. Remove the definitions and applications of the Virtual Chassis configuration groups.
3. Delete the preprovisioned member information configured at the **[edit virtual-chassis]** hierarchy level.
4. Delete any configured interfaces.
5. Remove the member IDs of each member router.

After you issue the **request virtual-chassis member-id delete** command on each router to remove the member ID, the router reboots and the software disables Virtual Chassis mode on that router.

Because the entire Virtual Chassis configuration is propagated from the master router to the other member router when the Virtual Chassis forms, you must delete each component of the Virtual Chassis configuration from both member routers, even though the component was originally configured only on the master router. For example, even though the preprovisioned member information was configured at the **[edit virtual-chassis]** hierarchy level only on the master router, you must delete the **virtual-chassis** stanza from the other member router in the Virtual Chassis.



NOTE: You cannot override a Virtual Chassis configuration simply by using the **load override** command to load a different configuration on the router from an ASCII file or from terminal input, as you can with other configurations. The member ID and Virtual Chassis port definitions are not stored in the configuration file, and are still defined even after the new configuration file is loaded.

This example deletes the Virtual Chassis configuration that uses the basic topology shown in [Figure 3 on page 75](#). For redundancy, each member router is configured with two Virtual Chassis ports, both of which must be removed as part of the deletion process.

Figure 3: Sample Topology for a Virtual Chassis with Two MX Series Routers

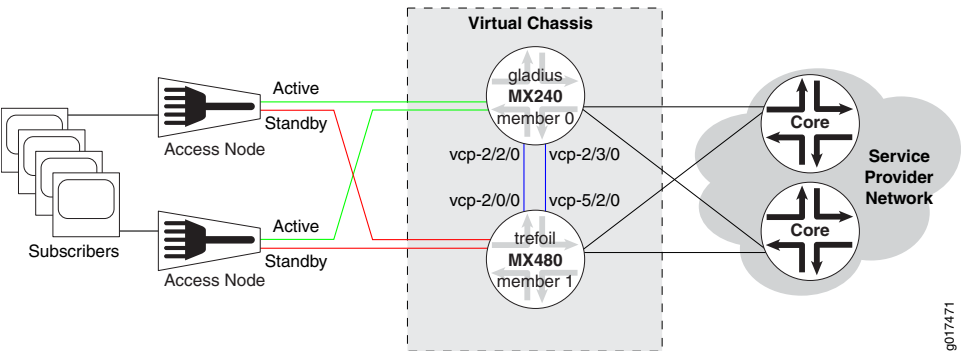


Table 9 on page 75 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 9: Components of the Sample MX Series Virtual Chassis

Router Name	Hardware	Serial Number	Member ID	Role	Virtual Chassis Ports	Network Port Slot Numbering
gladius	MX240 router with: <ul style="list-style-type: none">60-Gigabit Ethernet Enhanced Queuing MPC20-port Gigabit Ethernet MIC with SFP4-port 10-Gigabit Ethernet MIC with XFPMaster RE-S-2000 Routing Engine in slot 0 (represented in example as member0-re0)Backup RE-S-2000 Routing Engine in slot 1 (represented in example as member0-re1)	JN10C7135AFC	0	routing-engine (master)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 9: Components of the Sample MX Series Virtual Chassis (*continued*)

Router Name	Hardware	Serial Number	Member ID	Role	Virtual Chassis Ports	Network Port Slot Numbering
trefoil	MX480 router with: <ul style="list-style-type: none"> Two 30-Gigabit Ethernet Queuing MPCs Two 20-port Gigabit Ethernet MICs with SFP Two 2-port 10-Gigabit Ethernet MICs with XFP Master RE-S-2000 Routing Engine in slot 0 (represented in example as member1-re0) Backup RE-S-2000 Routing Engine in slot 1 (represented in example as member1-re1) 	JN115D117AFB	1	routing-engine (backup)	vcp-2/0/0 vcp-5/2/0	FPC 12 – 23 (offset = 12)

Configuration

To delete a Virtual Chassis configuration consisting of two MX Series routers, perform these tasks:

- [Deleting Virtual Chassis Ports on page 77](#)
- [Deleting Configuration Group Definitions and Applications on page 78](#)
- [Deleting Preprovisioned Member Information on page 80](#)
- [Deleting Configured Interfaces on page 80](#)
- [Deleting Member IDs to Disable Virtual Chassis Mode on page 82](#)

Deleting Virtual Chassis Ports

Step-by-Step Procedure To delete a Virtual Chassis port from a member router, you must use the **request virtual-chassis vc-port delete** command.



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

To remove the Virtual Chassis ports from each member router:

1. In the console window on member 0 (**gladius**), remove both Virtual Chassis ports (**vcp-2/2/0** and **vcp-2/3/0**).

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

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

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

```
user@gladius> request virtual-chassis vc-port delete fpc-slot 2 pic-slot 3 port 0
vc-port successfully deleted
```

2. In the console window on member 1 (**trefoil**), remove both Virtual Chassis ports (**vcp-2/0/0** and **vcp-5/2/0**).

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

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

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

```
user@trefoil> request virtual-chassis vc-port delete fpc-slot 5 pic-slot 2 port 0
vc-port successfully deleted
```

Results Display the results of the Virtual Chassis port deletion on each router. Confirm that no Virtual Chassis ports are listed in the output of either the **show virtual-chassis status** command or the **show virtual-chassis vc-port** command.

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

```
user@gladius> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: 4d6f.54cd.d2c1
```

				Mastership		Neighbor List	
Member ID	Status	Serial No	Model	priority	Role	ID	Interface
0 (FPC 0- 11)	Prsnt	JN10C7135AFC	mx240	129	Master*		
1 (FPC 12- 23)	NotPrsnt	JN115D117AFB	mx480				

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

```
user@gladius> show virtual-chassis vc-port  
member0:  
-----
```



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

Deleting Configuration Group Definitions and Applications

Step-by-Step Procedure

As part of deleting a Virtual Chassis configuration for MX Series routers with dual Routing Engines, you must delete the definitions and applications for the following configuration groups on both member routers:

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

To retain the information in these configuration groups before you delete them, you must copy them to the standard `re0` and `re1` configuration groups on the router, as described in the following procedure. For example, copy configuration groups `member0-re0` and `member1-re0` to `re0`, and copy `member0-re1` and `member1-re1` to `re1`.



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

To delete the configuration group definitions and applications for an MX Series Virtual Chassis:

1. In the console window on member 0 (**gladius**), delete the Virtual Chassis configuration group definitions and applications.
 - a. Copy the Virtual Chassis configuration groups to the standard configuration groups `re0` and `re1`.

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

```
user@gladius# copy groups member0-re0 to re0
user@gladius# copy groups member0-re1 to re1
```

- b. Apply the **re0** and **re1** configuration groups.

```
{master:member0-re0}[edit]
user@gladius# set apply-groups re0
user@gladius# set apply-groups re1
```

- c. Delete the Virtual Chassis configuration group definitions.

```
{master:member0-re0}[edit]
user@gladius# delete groups member0-re0
user@gladius# delete groups member0-re1
user@gladius# delete groups member1-re0
user@gladius# delete groups member1-re1
```

- d. Delete the Virtual Chassis configuration group applications.

```
{master:member0-re0}[edit]
user@gladius# delete apply-groups member0-re0
user@gladius# delete apply-groups member0-re1
user@gladius# delete apply-groups member1-re0
user@gladius# delete apply-groups member1-re1
```

2. In the console window on member 1 (**trefoil**), delete the Virtual Chassis configuration group definitions and applications.

- a. Copy the Virtual Chassis configuration groups to the standard configuration groups **re0** and **re1**.

```
{backup:member1-re0}[edit]
user@trefoil# copy groups member1-re0 to re0
user@trefoil# copy groups member1-re1 to re1
```

- b. Apply the **re0** and **re1** configuration groups.

```
{backup:member1-re0}[edit]
user@trefoil# set apply-groups re0
user@trefoil# set apply-groups re1
```

- c. Delete the Virtual Chassis configuration group definitions.

```
{backup:member1-re0}[edit]
user@trefoil# delete groups member0-re0
user@trefoil# delete groups member0-re1
user@trefoil# delete groups member1-re0
user@trefoil# delete groups member1-re1
```

- d. Delete the Virtual Chassis configuration group applications.

```
{backup:member1-re0}[edit]
user@trefoil# delete apply-groups member0-re0
user@trefoil# delete apply-groups member0-re1
user@trefoil# delete apply-groups member1-re0
user@trefoil# delete apply-groups member1-re1
```

Results Display the results of the configuration. Confirm that configuration groups **member0-re0**, **member 0-re1**, **member1-re0**, and **member1-re1** do not appear in the output of either the **show groups** command or the **show apply-groups** command.

```
[edit]
user@gladius# show groups ?

Possible completions:
<[Enter]>           Execute this command
<group_name>       Group name
global             Group name
re0                Group name
re1                Group name
|                  Pipe through a command

[edit]
user@gladius# show apply-groups
## Last changed: 2010-12-01 09:17:27 PST
apply-groups [ global re0 re1 ];
```

Deleting Preprovisioned Member Information

Step-by-Step Procedure You must delete the preprovisioned member information, which was configured at the `[edit virtual-chassis]` hierarchy level on the master router and then propagated to the backup router during the formation of the Virtual Chassis.

To delete the preprovisioned member information for the Virtual Chassis:

1. Delete the **virtual-chassis** configuration stanza on member 0 (**gladius**).

```
{master:member0-re0}[edit]
user@gladius# delete virtual-chassis
```

2. Delete the **virtual-chassis** configuration stanza on member 1 (**trefoil**).

```
{backup:member1-re0}[edit]
user@trefoil# delete virtual-chassis
```

Results Display the results of the deletion. Confirm that the **virtual-chassis** stanza no longer exists on either member router. For example, on **gladius** (member 0):

```
{master:member0-re0}[edit]
user@gladius# show virtual-chassis
<no output>
```

Deleting Configured Interfaces

Step-by-Step Procedure As part of deleting the Virtual Chassis, we recommend that you delete any interfaces that were configured when the Virtual Chassis was formed. This action ensures that nonexistent interfaces or interfaces belonging to the other member router do not remain on the router after Virtual Chassis mode is disabled.

To delete any interfaces that you configured when creating the Virtual Chassis:

1. In the console window on member 0 (**gladius**), delete any configured interfaces and commit the configuration.

- a. Delete the configured interfaces.

```
{master:member0-re0}[edit]
user@gladius# delete interfaces
```


- b. Commit the configuration on member 0.

```
{master:member0-re0}[edit system]
user@gladius# commit synchronize
member0-re0:
configuration check succeeds
member0-re1:
commit complete
member0-re0:
commit complete
```

2. In the console window on member 1 (**trefoil**), delete any configured interfaces and commit the configuration.

- a. Delete the configured interfaces.

```
{backup:member1-re0}[edit]
user@trefoil# delete interfaces
```

- b. Commit the configuration on member 1.

```
{backup:member1-re0}[edit system]
user@trefoil# commit synchronize
member1-re0:
configuration check succeeds
member1-re1:
commit complete
member1-re0:
commit complete
```



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

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

Deleting Member IDs to Disable Virtual Chassis Mode

Step-by-Step Procedure To delete a member ID from a Virtual Chassis member router, you must use the **request virtual-chassis member-id delete** command.



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

To delete the Virtual Chassis member IDs and disable Virtual Chassis mode:

1. In the console window on member 0 (**gladius**), delete the member ID and reboot the router.
 - a. Exit configuration mode.

```
{master:member0-re0}[edit]
user@gladius# exit
Exiting configuration mode
```
 - b. Delete member ID 0.

```
{master:member0-re0}

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

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

{master:member0-re0}
user@gladius>

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

System going down IMMEDIATELY
```
2. In the console window on member 1 (**trefoil**), delete the member ID and reboot the router.
 - a. Exit configuration mode.

```
{master:member1-re0}[edit]
user@trefoil# exit
Exiting configuration mode
```
 - b. Delete member ID 1.

```
{master:member1-re0}

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

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

```
{backup:member1-re0}
user@trefoil>

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

System going down IMMEDIATELY
```

Results After you issue the **request virtual-chassis member-id delete** command to remove the member ID, the router reboots and the software disables Virtual Chassis mode on that router. The routers that were formerly members of the Virtual Chassis now function as two independent routers.

Display the results of the configuration to confirm that the Virtual Chassis configuration has been deleted on each router. For example, on **gladius** (formerly member 0):

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

```
user@gladius> show virtual-chassis vc-port
error: the virtual-chassis-control subsystem is not running
```

Verification

To confirm that the Virtual Chassis configuration has been properly deleted, perform these tasks:

- [Verifying Deletion of the Virtual Chassis Ports on page 83](#)
- [Verifying Deletion of the Virtual Chassis Configuration Groups on page 84](#)
- [Verifying Deletion of the Virtual Chassis Member IDs on page 84](#)

Verifying Deletion of the Virtual Chassis Ports

Purpose Verify that the Virtual Chassis ports on both member routers have been deleted from the configuration.

Action Display the status of the Virtual Chassis configuration and Virtual Chassis ports.

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

```
user@gladius> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: 4d6f.54cd.d2c1
```

				Mastership		Neighbor List	
Member ID	Status	Serial No	Model	priority	Role	ID	Interface
0 (FPC 0- 11)	Prsnt	JN10C7135AFC	mx240	129	Master*		
1 (FPC 12- 23)	NotPrsnt	JN115D117AFB	mx480				

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

```
user@gladius> show virtual-chassis vc-port  
member0:  
-----
```

Meaning In the output of the **show virtual-chassis status** command, no Virtual Chassis ports (**vcp-slot/pic/port**) are displayed in the Neighbor List. The asterisk (*) following **Master** denotes the router on which the **show virtual-chassis status** command was issued.

In the output of the **show virtual-chassis vc-port** command, no Virtual Chassis ports are displayed on the router on which the command was issued.

Verifying Deletion of the Virtual Chassis Configuration Groups

Purpose Verify that the definitions and applications of the following Virtual Chassis configuration groups have been deleted from the global configuration:

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

Action Display the status of the Virtual Chassis configuration group definitions and applications.

```
[edit]  
user@gladius# show groups ?
```

Possible completions:

<[Enter]>	Execute this command
<group_name>	Group name
global	Group name
re0	Group name
re1	Group name
	Pipe through a command

```
[edit]  
user@gladius# show apply-groups  
apply-groups [ global re0 re1 ];
```

Meaning The output confirms that the Virtual Chassis configuration group definitions and applications have been deleted. In the output of both **show groups** and **show apply-groups**, only the standard configuration groups (**global**, **re0**, and **re1**) are listed. The Virtual Chassis configuration groups (**member0-re0**, **member0-re1**, **member1-re0**, and **member1-re1**) do not appear.

Verifying Deletion of the Virtual Chassis Member IDs

Purpose Verify that the member IDs for the Virtual Chassis have been deleted, and that the Virtual Chassis is no longer configured on either MX Series router.

Action Display the results of the configuration on each router. For example, on **trefoil** (formerly member 1):

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

```
user@trefoil> show virtual-chassis vc-port
error: the virtual-chassis-control subsystem is not running
```

Meaning When you attempt to issue either the **show virtual-chassis status** command or the **show virtual-chassis vc-port** command after the Virtual Chassis has been deleted, the router displays an error message indicating that the Virtual Chassis is no longer configured, and rejects the command.

- Related Documentation**
- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
 - [Virtual Chassis Components Overview on page 5](#)
 - [Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 34](#)

Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers

You can upgrade an MX Series Virtual Chassis configuration from Junos OS Release 11.2 to a later release. This upgrade procedure assumes that both member routers in the Virtual Chassis have dual Routing Engines installed.



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

This example describes how to upgrade Junos OS in a Virtual Chassis consisting of two MX Series routers, each of which has dual Routing Engines:

- [Requirements on page 85](#)
- [Overview and Topology on page 86](#)
- [Configuration on page 87](#)

Requirements

This example uses the following software and hardware and components:

- Junos OS Release 11.2
- One MX240 3D Universal Edge Router with dual Routing Engines
- One MX480 3D Universal Edge Router with dual Routing Engines

See [Table 10 on page 87](#) for information about the hardware installed in each MX Series router.

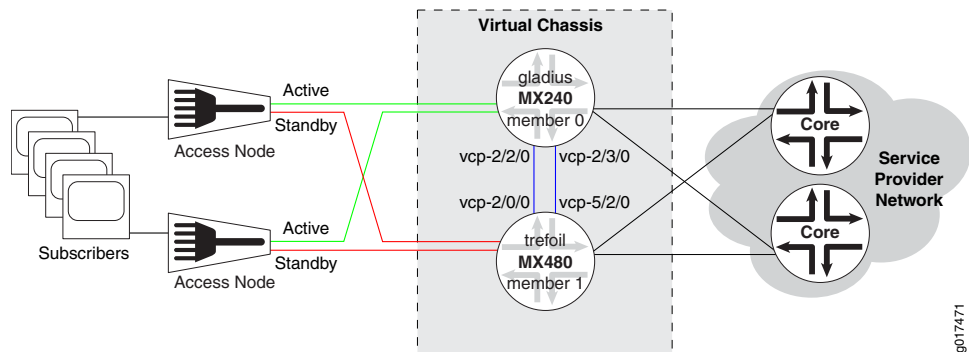
Overview and Topology

To upgrade Junos OS in an MX Series Virtual Chassis configuration, you must:

1. Prepare for the upgrade.
2. Install the Junos OS software package on each of the four Routing Engines.
3. Reboot the Routing Engines to run the new Junos OS release.
4. Re-enable graceful Routing Engine switchover and nonstop active routing.

This example upgrades Junos OS in an MX Series Virtual Chassis configuration that uses the basic topology shown in [Figure 4 on page 86](#). For redundancy, each member router is configured with two Virtual Chassis ports.

Figure 4: Sample Topology for a Virtual Chassis with Two MX Series Routers



[Table 10 on page 87](#) shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 10: Components of the Sample MX Series Virtual Chassis

Router Name	Hardware	Serial Number	Member ID	Role	Virtual Chassis Ports	Network Port Slot Numbering
gladius	MX240 router with: <ul style="list-style-type: none"> • 60-Gigabit Ethernet Enhanced Queuing MPC • 20-port Gigabit Ethernet MIC with SFP • 4-port 10-Gigabit Ethernet MIC with XFP • Master RE-S-2000 Routing Engine in slot 0 (represented in example as member0-re0) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as member0-re1) 	JN10C7135AFC	0	routing-engine (master)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11
trefoil	MX480 router with: <ul style="list-style-type: none"> • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Master RE-S-2000 Routing Engine in slot 0 (represented in example as member1-re0) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as member1-re1) 	JN115D117AFB	1	routing-engine (backup)	vcp-2/0/0 vcp-5/2/0	FPC 12 – 23 (offset = 12)

Configuration

To upgrade Junos OS in a Virtual Chassis configuration consisting of two MX Series routers, each with dual Routing Engines, perform these tasks:

- [Preparing for the Upgrade on page 88](#)
- [Installing the Junos OS Software Package on Each Routing Engine on page 88](#)

- [Rebooting the Routing Engines to Run the New Junos OS Release on page 89](#)
- [Re-enabling Graceful Routing Engine Switchover and Nonstop Active Routing on page 89](#)

Preparing for the Upgrade

Step-by-Step Procedure

To prepare for the upgrade process:

1. Use FTP or a Web browser to download the Junos OS software package to the master Routing Engine on the Virtual Chassis master router (**member0-re0**).
See Downloading Software in the Installation and Upgrade Guide.
2. Disable nonstop active routing on the master router.

```
{master:member0-re0}[edit routing-options]  
user@gladius# delete nonstop-routing
```
3. Disable graceful Routing Engine switchover on the master router.

```
{master:member0-re0}[edit chassis redundancy]  
user@gladius# delete graceful-switchover
```
4. Commit the configuration on the master router.

```
{master:member0-re0}[edit system]  
user@gladius# commit synchronize
```
5. Exit CLI configuration mode.

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

Installing the Junos OS Software Package on Each Routing Engine

Step-by-Step Procedure

Installing the Junos OS software package on each Routing Engine in an MX Series Virtual Chassis prepares the Routing Engines to run the new software release after a reboot. This action is also referred to as *arming* the Routing Engines.

To install the Junos OS software package on all four Routing Engines from the master router (**member0-re0**) in the Virtual Chassis:

1. Install the software package on **member0-re0**. This command also propagates the software package to **member1-re0**.

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

For example:

```
{master:member0-re0}  
user@gladius> request system software add jinstall-11.2R1-8-domestic-signed.tgz
```
2. Install the software package on **member0-re1**.

```
{master:member0-re0}  
user@gladius> request system software add package-name re1
```
3. Install the software package on **member1-re1**.

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



```
user@gladius> request system software add package-name member 1 re1
```

Results Display the results of the installation. Verify that the correct software package has been installed on the local master Routing Engine in **member 0** (**member0-re0**) and on the local master Routing Engine in **member 1** (**member1-re0**).

```
{master:member0-re0}

user@gladius> show version
member0:
-----
Hostname: gladius
Model: mx240
. . .
JUNOS Installation Software [11.2R1-8]

member1:
-----
Hostname: trefoil
Model: mx480
. . .
JUNOS Installation Software [11.2R1-8]
```

Rebooting the Routing Engines to Run the New Junos OS Release

Step-by-Step Procedure To reboot each of the four Routing Engines in an MX Series Virtual Chassis from the Virtual Chassis master router (**member0-re0**):

1. Reboot **member1-re1**.

```
{master:member0-re0}

user@gladius> request system reboot member 1 other-routing-engine
```
2. Reboot **member0-re1**.

```
{master:member0-re0}

user@gladius> request system reboot other-routing-engine
```
3. Reboot both master Routing Engines (**member0-re0** and **member1-re0**).

```
{master:member0-re0}

user@gladius> request system reboot all-members
```

This command reboots all line cards in **member 0** (**gladius**) and **member 1** (**trefoil**) to use the new Junos OS release. A traffic disruption occurs until all line cards are back online and the Virtual Chassis re-forms.

Re-enabling Graceful Routing Engine Switchover and Nonstop Active Routing

Step-by-Step Procedure After upgrading the Junos OS release, you need to re-enable graceful Routing Engine switchover and nonstop active routing for the Virtual Chassis.

To re-enable graceful Routing Engine switchover and nonstop active routing from the Virtual Chassis master router (**member0-re0**):

1. In the console window on **member 0** (**gladius**), enable graceful Routing Engine switchover on the master router.

```
{master:member0-re0}[edit chassis redundancy]
user@gladius# set graceful-switchover
2. Re-enable nonstop active routing on the master router.
{master:member0-re0}[edit routing-options]
user@gladius# set nonstop-routing
3. Commit the configuration on the master router.
{master:member0-re0}[edit system]
user@gladius# commit synchronize
```

**Related
Documentation**

- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
- [Virtual Chassis Components Overview on page 5](#)
- [Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 35](#)

Example: Replacing a Routing Engine in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers

If you remove a Routing Engine from a member router in an MX Series Virtual Chassis for upgrade or repair, you must replace it with a new Routing Engine in the empty Routing Engine slot, and install the same Junos OS release on the new Routing Engine that is running on the other Routing Engines in the Virtual Chassis. The Virtual Chassis remains operational during the replacement procedure.

All four Routing Engines (both Routing Engines in the master router and both Routing Engines in the backup router) in the Virtual Chassis must run the same Junos OS release.



BEST PRACTICE: We recommend that you replace a Routing Engine in an MX Series Virtual Chassis configuration during a maintenance window to minimize the possibility of disruption to subscribers.

This example describes how to replace a Routing Engine in an MX Series Virtual Chassis configuration consisting of two MX Series routers, each of which has dual Routing Engines installed:

- [Requirements on page 90](#)
- [Overview and Topology on page 91](#)
- [Configuration on page 92](#)
- [Verification on page 96](#)

Requirements

This example uses the following software and hardware and components:

- Junos OS Release 11.4

- One MX240 3D Universal Edge Router with dual Routing Engines
- One MX480 3D Universal Edge Router with dual Routing Engines

See [Table 11 on page 92](#) for information about the hardware installed in each MX Series router.

Overview and Topology

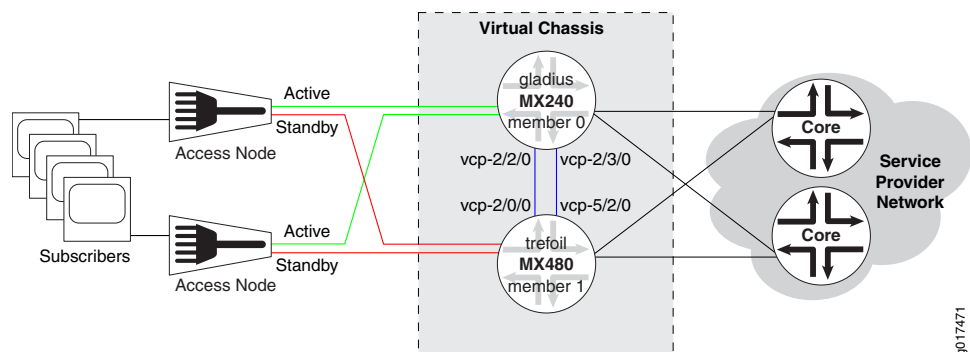
To replace a Routing Engine in an MX Series Virtual Chassis configuration, you must:

1. Remove the Routing Engine that needs repair or upgrade.
2. Return the Routing Engine to Juniper Networks, Inc.
3. Install the new Routing Engine in the empty Routing Engine slot.
4. install the same Junos OS release on the new Routing Engine that is running on the other Routing Engines in the Virtual Chassis.
5. Reboot the new Routing Engine to run the Junos OS software release.

[Figure 5 on page 91](#) shows the topology of the MX Series Virtual Chassis configuration used in this example. This example replaces the backup RE-S-2000 Routing Engine in slot 1 of the Virtual Chassis backup router, which is an MX480 router named **trefoil** that is assigned member ID 1. The backup Routing Engine in slot 1 of **trefoil** is represented in the example as **member1-re1**.

For redundancy, each of the two member routers is configured with two Virtual Chassis ports.

Figure 5: Sample Topology for a Virtual Chassis with Two MX Series Routers



[Table 11 on page 92](#) shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 11: Components of the Sample MX Series Virtual Chassis

Router Name	Hardware	Serial Number	Member ID	Role	Virtual Chassis Ports	Network Port Slot Numbering
gladius	MX240 router with: <ul style="list-style-type: none"> • 60-Gigabit Ethernet Enhanced Queuing MPC • 20-port Gigabit Ethernet MIC with SFP • 4-port 10-Gigabit Ethernet MIC with XFP • Master RE-S-2000 Routing Engine in slot 0 (represented in example as member0-re0) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as member0-re1) 	JN10C7135AFC	0	routing-engine (master)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11
trefoil	MX480 router with: <ul style="list-style-type: none"> • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Master RE-S-2000 Routing Engine in slot 0 (represented in example as member1-re0) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as member1-re1) 	JN115D117AFB	1	routing-engine (backup)	vcp-2/0/0 vcp-5/2/0	FPC 12 – 23 (offset = 12)

Configuration

To replace a Routing Engine in a Virtual Chassis configuration consisting of two MX Series routers, each with dual Routing Engines, perform these tasks:

- [Removing the Routing Engine on page 94](#)
- [Returning the Routing Engine to Juniper Networks, Inc. on page 94](#)

- [Installing the New Routing Engine on page 94](#)
- [Installing the Junos OS Release on the New Routing Engine on page 94](#)

Removing the Routing Engine

Step-by-Step Procedure

To remove the Routing Engine that needs repair or upgrade:

- Remove the Routing Engine according to the procedure for your MX Series router.
 - For an MX240 router, see Removing an MX240 Routing Engine in the [MX240 3D Universal Edge Router Hardware Guide](#).
 - For an MX480 router, see Removing an MX480 Routing Engine in the [MX480 3D Universal Edge Router Hardware Guide](#).
 - For an MX960 router, see Removing an MX960 Routing Engine in the [MX960 3D Universal Edge Router Hardware Guide](#).

Returning the Routing Engine to Juniper Networks, Inc.

Step-by-Step Procedure

To return the Routing Engine to Juniper Networks, Inc:

- Obtain a Return Materials Authorization (RMA) from the Juniper Networks Technical Assistance Center (JTAC) and return the Routing Engine to Juniper Networks, Inc.

For instructions, see Returning a Hardware Component to Juniper Networks, Inc. in the *Hardware Guide* for your MX Series router.

Installing the New Routing Engine

Step-by-Step Procedure

To install the new Routing Engine in the Virtual Chassis member router:

- Install the Routing Engine in the empty Routing Engine slot of the member router according to the procedure for your MX Series router.
 - For an MX240 router, see Installing an MX240 Routing Engine in the [MX240 3D Universal Edge Router Hardware Guide](#).
 - For an MX480 router, see Installing an MX480 Routing Engine in the [MX480 3D Universal Edge Router Hardware Guide](#).
 - For an MX960 router, see Installing an MX960 Routing Engine in the [MX960 3D Universal Edge Router Hardware Guide](#).

Installing the Junos OS Release on the New Routing Engine

Step-by-Step Procedure

You must install the same Junos OS release on the new Routing Engine that is running on the other Routing Engines in the MX Series Virtual Chassis. Installing the Junos OS software prepares the Routing Engine to run the new Junos OS release after a reboot. This action is also referred to as *arming* the Routing Engine.

To install the Junos OS release on the new Routing Engine (**member1-re1**) in the Virtual Chassis:

1. Use FTP or a Web browser to download the Junos OS software to the master Routing Engine on the Virtual Chassis master router (**member0-re0**).

See Downloading Software in the Installation and Upgrade Guide.



NOTE: Make sure you download and install the same Junos OS release that is running on all Routing Engines in the Virtual Chassis.

2. Log in to the console of the new Routing Engine as the user **root** with no password.
3. From the console, configure a plain-text password for the **root** (superuser) login.

```
{local:member1-re1}[edit system]
root# set root-authentication plain-text-password
New password: type password here
Retype new password: retype password here
```

4. From the console, commit the configuration.

```
{local:member1-re1}[edit]
root# commit synchronize and-quit
...
member1-re0:
configuration check succeeds
member0-re0:
commit complete
member1-re0:
commit complete
member1-re1:
commit complete
Exiting configuration mode
```

5. Use Telnet or SSH to log in to the member router containing the new Routing Engine (**trefoil**).

```
{local:member1-re1}
user@trefoil>
```

Notice that the router name (**trefoil**) now appears in the command prompt.

6. Install the Junos OS release on the new Routing Engine (**member1-re1**) from the Virtual Chassis master router (**member0-re0**).

```
{master:member0-re0}
user@trefoil> request system software add member member-id re1 no-validate reboot
package-name force
```

For example:

```
{master:member0-re0}
user@trefoil> request system software add member 1 re1 no-validate reboot
/var/tmp/jinstall-11.4R1-8-domestic-signed.tgz force
Pushing bundle to re1...
```

This command reboots **member1-re1** after the software is added.

Results After the reboot, the new Routing Engine becomes part of the Virtual Chassis, updates its command prompt to display **member1-re1**, and copies the appropriate configuration from the Virtual Chassis.

Verification

To verify that the MX Series Virtual Chassis is operating properly with the new Routing Engine, perform these tasks:

- [Verifying the Junos OS Installation on the New Routing Engine on page 96](#)
- [Verifying the Junos OS License Installation on the New Routing Engine on page 96](#)
- [Switching the Local Mastership in the Member Router to the New Routing Engine on page 97](#)

Verifying the Junos OS Installation on the New Routing Engine

Purpose Verify that you have installed the correct Junos OS release on the new Routing Engine (**member1-re1**).

Action Display the hostname, model name, and version information of the Junos OS release running on the new Routing Engine.

```
{local:member1-re1}
user@trefoil> show version local
Hostname: trefoil
Model: mx480
. . .
JUNOS Base OS boot [11.4R1-8]
JUNOS Base OS Software Suite [11.4R1-8]
. . .
```

Meaning The relevant portion of the **show version local** command output confirms that Junos OS Release 11.4R1-8 was installed as intended.

Verifying the Junos OS License Installation on the New Routing Engine

Purpose Verify that the MX Virtual Chassis Redundancy Feature Pack and the required Junos OS feature licenses are properly installed on the member router containing the new Routing Engine.

For information about license installation, see:

- [Installing Junos OS Licenses on Virtual Chassis Member Routers on page 40](#)
- [Software Features That Require Licenses on MX Series Routers Only](#)

Action Display the Junos OS licenses installed on the new Routing Engine.

```
{local:member1-re1}
user@trefoil> show system license
License usage:
                Licenses      Licenses      Licenses      Expiry
```


Feature name	used	installed	needed	
subscriber-accounting	0	1	0	permanent
subscriber-authentication	0	1	0	permanent
subscriber-address-assignment	0	1	0	permanent
subscriber-vlan	0	1	0	permanent
subscriber-ip	0	1	0	permanent
scale-subscriber	0	256000	0	permanent
scale-l2tp	0	1000	0	permanent
scale-mobile-ip	0	1000	0	permanent
virtual-chassis	0	1	0	permanent

Meaning The `show system license` command output confirms that the MX Virtual Chassis Redundancy Feature Pack has been installed on this member router. In addition, the necessary Junos OS feature licenses have been installed to enable use of a particular software feature or scaling level.

Switching the Local Mastership in the Member Router to the New Routing Engine

Purpose Verify that the MX Series Virtual Chassis is operating properly with the new Routing Engine by confirming that the new Routing Engine can take over local mastership from the existing Routing Engine in the Virtual Chassis backup router, **trefoil** (member 1).

Action Switch the local mastership of the Routing Engines in **trefoil** from the Routing Engine in slot 0 (**member1-re0**) to the newly installed Routing Engine in slot 1 (**member1-re1**).

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

```
user@trefoil> request chassis routing-engine master switch
```

Wait approximately 1 minute to display the status and roles of the member routers in the Virtual Chassis after the local switchover.

```
{backup:member1-re1}
```

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

Preprovisioned Virtual Chassis

Virtual Chassis ID: a5b6.be0c.9525

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID	Interface
0 (FPC 0- 11)	Prsnt	JN10C7135AFC	mx240	129	Master	1	vcp-2/2/0
						1	vcp-2/3/0
1 (FPC 12- 23)	Prsnt	JN115D117AFB	mx480	129	Backup*	0	vcp-2/0/0
						0	vcp-5/2/0

Meaning Issuing the `request chassis routing-engine master switch` command to initiate the local switchover of the Routing Engines in the Virtual Chassis backup router (**trefoil**) affects only the roles of the Routing Engines in that member router (**member1-re0** and **member1-re1**), but does not change the global mastership of the Virtual Chassis. The output of the `show virtual-chassis status` command confirms that after the local switchover, member 0 (**gladius**) is still the Virtual Chassis master router, and member 1 (**trefoil**) is still the Virtual Chassis backup router.

Before the local switchover, **member1-re0** was the master Routing Engine in the Virtual Chassis backup router (VC-Bm), and **member1-re1** (the new Routing Engine) was the standby Routing Engine in the Virtual Chassis backup router (VC-Bs).

After the local switchover, **member1-re0** and **member1-re1** switch roles. The new Routing Engine, **member1-re1**, becomes the master Routing Engine in the Virtual Chassis backup router (VC-Bm), and **member1-re0** becomes the standby Routing Engine in the Virtual Chassis backup router (VC-Bs).

Table 12 on page 98 lists the role transitions that occur for each member router and Routing Engine before and after the local switchover of the Routing Engines in **trefoil**.



NOTE: The role transitions described in Table 12 on page 98 apply only when you initiate the local switchover from the Virtual Chassis backup router (VC-B). For information about the role transitions that occur when you initiate the local switchover from the Virtual Chassis master router (VC-M), see “Switchover Behavior in a Virtual Chassis” on page 17.

Table 12: Virtual Chassis Role Transitions Before and After Local Routing Engine Switchover

Virtual Chassis Component	Role <i>Before</i> Local Switchover	Role <i>After</i> Local Switchover
gladius (member 0)	Virtual Chassis master router (VC-M)	Virtual Chassis master router (VC-M)
trefoil (member 1)	Virtual Chassis backup router (VC-B)	Virtual Chassis backup router (VC-B)
member0-re0	Master Routing Engine in the Virtual Chassis master router (VC-Mm)	Master Routing Engine in the Virtual Chassis master router (VC-Mm)
member0-re1	Standby Routing Engine in the Virtual Chassis master router (VC-Ms)	Standby Routing Engine in the Virtual Chassis master router (VC-Ms)
member1-re0	Master Routing Engine in the Virtual Chassis backup router (VC-Bm)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)
member1-re1 (new Routing Engine)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Master Routing Engine in the Virtual Chassis backup router (VC-Bm)



BEST PRACTICE: After you switch the local mastership of the Routing Engines, full synchronization of the Routing Engines takes approximately 30 minutes to complete. To prevent possible loss of subscriber state information due to

incomplete synchronization, we recommend that you wait at least 30 minutes before performing another local switchover, global switchover, or graceful Routing Engine switchover in an MX Series Virtual Chassis configuration.

Related Documentation

- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
- [Virtual Chassis Components Overview on page 5](#)
- [Installing Junos OS Licenses on Virtual Chassis Member Routers on page 40](#)
- [Switchover Behavior in a Virtual Chassis on page 17](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)
- [Installation and Upgrade Guide](#)

Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers

This example illustrates a typical class of service (CoS) configuration that you might want to use for the Virtual Chassis ports in an MX Series Virtual Chassis.

- [Requirements on page 99](#)
- [Overview on page 99](#)
- [Configuration on page 100](#)

Requirements

Before you begin:

- Configure a Virtual Chassis consisting of two MX Series routers.

See “[Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis](#)” on page 59

Overview

By default, all Virtual Chassis ports in an MX Series Virtual Chassis use a default CoS configuration specifically tailored for Virtual Chassis ports. The default configuration, which applies to all Virtual Chassis ports in the Virtual Chassis, includes classifiers, forwarding classes, rewrite rules, and schedulers. This default CoS configuration prioritizes internal Virtual Chassis Control Protocol (VCCP) traffic that traverses the Virtual Chassis port interfaces, and prioritizes control traffic over user traffic on the Virtual Chassis ports. In most cases, the default CoS configuration is adequate for your needs without requiring any additional CoS configuration.

In some cases, however, you might want to customize the traffic-control profile configuration on Virtual Chassis ports. For example, you might want to assign different priorities and excess rates to different forwarding classes. To create a nondefault CoS

configuration, you can create an output traffic-control profile that defines a set of traffic scheduling resources and references a scheduler map. You then apply the output traffic-control profile to all Virtual Chassis port interfaces at once by using **vcp-*** as the interface name representing all Virtual Chassis ports. You cannot configure CoS for Virtual Chassis ports on an individual basis.

Table 13 on page 100 shows the nondefault CoS scheduler hierarchy configured in this example for the Virtual Chassis ports.

Table 13: Sample CoS Scheduler Hierarchy for Virtual Chassis Ports

Traffic Type	Queue Number	Priority	Transmit Rate/ Excess Rate
Network control (VCCP traffic)	3	Medium	90%
Expedited forwarding (voice traffic)	2	High	10%
Assured forwarding (video traffic)	1	Excess Low	99%
Best effort (data traffic)	0	Excess Low	1%

In this example, you create a nondefault CoS configuration for Virtual Chassis ports by completing the following tasks on the Virtual Chassis master router:

- Associate forwarding classes with **queue 0** through **queue 3**, and configure a fabric priority value for each queue.
- Configure an output traffic control profile named **tcp-vcp-ifd** to define traffic scheduling parameters, and associate a scheduler map named **sm-vcp-ifd** with the traffic control profile.
- Apply the output traffic-control profile to the **vcp-*** interface, which represents all Virtual Chassis port interfaces in the Virtual Chassis.
- Associate the **sm-vcp-ifd** scheduler map with the forwarding classes and scheduler configuration.
- Configure the parameters for schedulers **s-medium-priority**, **s-high-priority**, **s-low-priority**, **s-high-weight**, and **s-low-weight**.

Configuration

CLI Quick Configuration

To quickly create a nondefault CoS configuration for Virtual Chassis ports, copy the following commands and paste them into the router terminal window:

```
[edit]
set class-of-service forwarding-classes queue 0 best-effort
set class-of-service forwarding-classes queue 0 priority low
set class-of-service forwarding-classes queue 1 assured-forwarding
set class-of-service forwarding-classes queue 1 priority low
```

```

set class-of-service forwarding-classes queue 2 expedited-forwarding
set class-of-service forwarding-classes queue 2 priority high
set class-of-service forwarding-classes queue 3 network-control
set class-of-service forwarding-classes queue 3 priority high
set class-of-service traffic-control-profiles tcp-vcp-ifd scheduler-map sm-vcp-ifd
set class-of-service interfaces vcp-* output-traffic-control-profile tcp-vcp-ifd
set class-of-service scheduler-maps sm-vcp-ifd forwarding-class network-control
  scheduler s-medium-priority
set class-of-service scheduler-maps sm-vcp-ifd forwarding-class expedited-forwarding
  scheduler s-high-priority
set class-of-service scheduler-maps sm-vcp-ifd forwarding-class assured-forwarding
  scheduler s-high-weight
set class-of-service scheduler-maps sm-vcp-ifd forwarding-class best-effort scheduler
  s-low-weight
set class-of-service schedulers s-medium-priority transmit-rate percent 10
set class-of-service schedulers s-medium-priority priority medium-high
set class-of-service schedulers s-medium-priority excess-priority high
set class-of-service schedulers s-high-priority transmit-rate percent 90
set class-of-service schedulers s-high-priority priority high
set class-of-service schedulers s-high-priority excess-priority high
set class-of-service schedulers s-low-priority priority low
set class-of-service schedulers s-high-weight excess-rate percent 99
set class-of-service schedulers s-low-weight excess-rate percent 1

```

Step-by-Step Procedure To create a nondefault CoS configuration for Virtual Chassis ports in an MX Series Virtual Chassis:

1. Log in to the console on the master router of the Virtual Chassis.
2. Specify that you want to configure CoS forwarding classes.


```
{master:member0-re0} [edit]
user@host# edit class-of-service forwarding-classes
```
3. Associate a forwarding class with each queue name and number, and configure a fabric priority value for each queue.


```
{master:member0-re0} [edit class-of-service forwarding-classes]
user@host# set queue 0 best-effort priority low
user@host# set queue 1 assured-forwarding priority low
user@host# set queue 2 expedited-forwarding priority high
user@host# set queue 3 network-control priority high
```
4. Return to the **[edit class-of-service]** hierarchy level to configure an output traffic-control profile.


```
{master:member0-re0} [edit class-of-service forwarding-classes]
user@host# up
```
5. Configure an output traffic-control profile and associate it with a scheduler map.


```
{master:member0-re0} [edit class-of-service]
user@host# set traffic-control-profiles tcp-vcp-ifd scheduler-map sm-vcp-ifd
```
6. Apply the output traffic-control profile to all Virtual Chassis port interfaces in the Virtual Chassis.


```
{master:member0-re0} [edit class-of-service]
user@host# set interfaces vcp-* output-traffic-control-profile tcp-vcp-ifd
```

7. Specify that you want to configure the scheduler map.

```
{master:member0-re0} [edit class-of-service]
user@host# edit scheduler-maps sm-vcp-ifd
```
8. Associate the scheduler map with the scheduler configuration and forwarding classes.

```
{master:member0-re0} [edit class-of-service scheduler-maps sm-vcp-ifd]
user@host# set forwarding-class network-control scheduler s-medium-priority
user@host# set forwarding-class expedited-forwarding scheduler s-high-priority
user@host# set forwarding-class assured-forwarding scheduler s-high-weight
user@host# set forwarding-class best-effort scheduler s-low-weight
```
9. Return to the **[edit class-of-service]** hierarchy level to configure the schedulers.

```
{master:member0-re0} [edit class-of-service scheduler-maps sm-vcp-ifd]
user@host# up 2
```
10. Configure parameters for the schedulers.

```
{master:member0-re0} [edit class-of-service]
user@host# set schedulers s-medium-priority priority medium-high excess-priority
high transmit-rate percent 10
user@host# set schedulers s-high-priority priority high excess-priority high
transmit-rate percent 90
user@host# set schedulers s-low-priority priority low
user@host# set schedulers s-high-weight excess-rate percent 99
user@host# set schedulers s-low-weight excess-rate percent 1
```

Results From the **[edit class-of-service]** hierarchy level in configuration mode, confirm the results of your configuration by issuing the **show** statement. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
{master:member0-re0} [edit class-of-service]
user@host# show
forwarding-classes {
  queue 0 best-effort priority low;
  queue 1 assured-forwarding priority low;
  queue 2 expedited-forwarding priority high;
  queue 3 network-control priority high;
}
traffic-control-profiles {
  tcp-vcp-ifd {
    scheduler-map sm-vcp-ifd;
  }
}
interfaces {
  vcp-* {
    output-traffic-control-profile tcp-vcp-ifd;
  }
}
scheduler-maps {
  sm-vcp-ifd {
    forwarding-class network-control scheduler s-medium-priority;
    forwarding-class expedited-forwarding scheduler s-high-priority;
```

```
        forwarding-class assured-forwarding scheduler s-high-weight;
        forwarding-class best-effort scheduler s-low-weight;
    }
}
schedulers {
    s-medium-priority {
        transmit-rate percent 10;
        priority medium-high;
        excess-priority high;
    }
    s-high-priority {
        transmit-rate percent 90;
        priority high;
        excess-priority high;
    }
    s-low-priority {
        priority low;
    }
    s-high-weight {
        excess-rate percent 99;
    }
    s-low-weight {
        excess-rate percent 1;
    }
}
```

If you are done configuring CoS on the master router, enter **commit** from configuration mode.

**Related
Documentation**

- [Class of Service Overview for Virtual Chassis Ports on page 25](#)
- [Guidelines for Configuring Class of Service for Virtual Chassis Ports on page 30](#)
- Junos OS Class of Service Configuration Guide

CHAPTER 7

Configuration Statements

member (MX Series Virtual Chassis)

Syntax	<code>member <i>member-id</i> { <i>role</i> (routing-engine line-card); <i>serial-number</i> <i>serial-number</i>; }</code>
Hierarchy Level	[edit <code>virtual-chassis</code>]
Release Information	Statement introduced in Junos OS Release 11.2.
Description	Configure an MX Series router as a member of a Virtual Chassis configuration. You can configure a maximum of two member routers in an MX Series Virtual Chassis.
Options	<i>member-id</i> —Numeric value that identifies a member router in a Virtual Chassis configuration. Values: 0 or 1 The remaining statements are explained separately.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring Preprovisioned Member Information for a Virtual Chassis on page 44• Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59

no-split-detection (MX Series Virtual Chassis)

Syntax	no-split-detection;
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 11.2.
Description	As part of the preprovisioned configuration for an MX Series Virtual Chassis, disable detection of a split in the Virtual Chassis configuration. By default, split detection in the Virtual Chassis is enabled. To maintain the Virtual Chassis configuration in the event of a failure of one of the two member routers, we recommend that you use the no-split-detection statement to disable split detection in Virtual Chassis configurations in which you think the backup router is more likely to fail than the link to the backup router.



BEST PRACTICE: We recommend that you use the **no-split-detection** statement to disable split detection for a two-member MX Series Virtual Chassis configuration if you think the backup router is more likely to fail than the Virtual Chassis port links to the backup router. Configuring redundant Virtual Chassis ports on different line cards in each member router reduces the likelihood that all Virtual Chassis port interfaces to the backup router can fail.

Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring Preprovisioned Member Information for a Virtual Chassis on page 44• Disabling Split Detection in a Virtual Chassis Configuration on page 57• Split Detection Behavior in a Virtual Chassis on page 19• Global Roles and Local Roles in a Virtual Chassis on page 13• Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59

preprovisioned (MX Series Virtual Chassis)

Syntax	preprovisioned;
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 11.2.
Description	<p>Enable creation of a Virtual Chassis by means of a preprovisioned configuration.</p> <p>To configure a Virtual Chassis consisting of MX Series routers, you must create a preprovisioned configuration on the master router in the Virtual Chassis by specifying the serial number, member ID, and role for each router (member chassis) in the Virtual Chassis. When a new member router joins the Virtual Chassis, its serial number is compared against the values specified in the preprovisioned configuration. If the serial number of a joining router does not match any of the configured serial numbers, the software prevents that router from becoming a member of the Virtual Chassis.</p>
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring Preprovisioned Member Information for a Virtual Chassis on page 44• Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59

role (MX Series Virtual Chassis)

Syntax	role (routing-engine line-card);
Hierarchy Level	[edit virtual-chassis member <i>member-id</i>]
Release Information	Statement introduced in Junos OS Release 11.2.
Description	As part of the preprovisioned configuration for an MX Series Virtual Chassis, assign the role to be performed by each member router in the Virtual Chassis. The preprovisioned configuration permanently associates the member ID and role with the member router's chassis serial number.
Options	<p>routing-engine—Enable the member router to function as the master router or backup router of the Virtual Chassis configuration. The master router maintains the global configuration and state information for both members of the Virtual Chassis, and runs the chassis management processes. The backup router synchronizes with the master router and relays chassis control information, such as line-card presence and alarms, to the master router. If the master router is unavailable, the backup router takes mastership of the Virtual Chassis to preserve routing information and maintain network connectivity without disruption. You must assign the routing-engine role to both members of the Virtual Chassis. When the Virtual Chassis is formed, the software runs a mastership election algorithm to determine which of the two member routers functions as the master router and which functions as the backup router of the Virtual Chassis.</p> <p>line-card—Explicitly configuring a member router with the line-card role is <i>not supported</i> in the current release. However, when split detection is enabled (the default behavior for a Virtual Chassis) and either the Virtual Chassis ports go down or the backup router fails, the master router takes a line-card role. The line-card role effectively removes the former master router from the Virtual Chassis configuration until connectivity is restored.</p>
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring Preprovisioned Member Information for a Virtual Chassis on page 44• Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59• Virtual Chassis Components Overview on page 5• Global Roles and Local Roles in a Virtual Chassis on page 13• Split Detection Behavior in a Virtual Chassis on page 19

serial-number (MX Series Virtual Chassis)

Syntax	<code>serial-number <i>serial-number</i>;</code>
Hierarchy Level	[edit virtual-chassis member <i>member-id</i>]
Release Information	Statement introduced in Junos OS Release 11.2.
Description	As part of the preprovisioned configuration for an MX Series Virtual Chassis, specify the chassis serial number of each MX Series member router in the Virtual Chassis configuration. If you do not correctly specify a router's serial number in the preprovisioned configuration, the software does not recognize that router as a member of the Virtual Chassis.
Options	<i>serial-number</i> —Alphanumeric string that represents the chassis serial number of each member router in the Virtual Chassis configuration. The chassis serial number is located on a label affixed to the side of the MX Series chassis. You can also obtain the router's chassis serial number by issuing the show chassis hardware command.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Configuring Preprovisioned Member Information for a Virtual Chassis on page 44 • Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59

tracoptions (MX Series Virtual Chassis)

Syntax	<pre>tracoptions { file <i>filename</i> <files <i>number</i>> <match <i>regular-expression</i>> <no-stamp> <replace> <size <i>maximum-file-size</i>> <world-readable no-world-readable>; flag <i>flag</i> <detail> <disable> <receive> <send>; }</pre>
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 11.2.
Description	Define tracing operations for the MX Series Virtual Chassis configuration.
Default	Tracing operations are disabled.
Options	<p>detail—(Optional) Generate detailed trace information for a flag.</p> <p>disable—(Optional) Disable a flag.</p> <p>file <i>filename</i>—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 <code>/var/log</code>.</p> <p>files <i>number</i>—(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.</p> <p>Range: 2 through 1000</p> <p>Default: 3 files</p> <p>flag <i>flag</i>—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:</p> <ul style="list-style-type: none">• all—All tracing operations.• auto-configuration—Trace Virtual Chassis ports that have been automatically configured.• csn—Trace Virtual Chassis complete sequence number (CSN) packets.• error—Trace Virtual Chassis errored packets.• graceful-restart—Trace Virtual Chassis graceful restart events.• hello—Trace Virtual Chassis hello packets.• krt—Trace Virtual Chassis kernel routing table (KRT) events.• lsp—Trace Virtual Chassis link-state packets.• lsp-generation—Trace Virtual Chassis link-state packet generation.• me—Trace Virtual Chassis mastership election (ME) events.

- **normal**—Trace normal events.
- **packets**—Trace Virtual Chassis packets.
- **parse**—Trace reading of the configuration.
- **psn**—Trace partial sequence number (PSN) packets.
- **route**—Trace Virtual Chassis routing information.
- **spf**—Trace Virtual Chassis shortest-path-first (SPF) events.
- **state**—Trace Virtual Chassis state transitions.
- **task**—Trace Virtual Chassis task operations.

match *regular-expression*—(Optional) Refine the output to include lines that contain the regular expression.

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 instead of appending information to it.

send—(Optional) Trace transmitted packets.

size *maximum-file-size*—(Optional) Maximum size of each trace file. By default, the number entered is treated as bytes. Alternatively, you can include a suffix to the number to indicate kilobytes (KB), megabytes (MB), or gigabytes (GB). If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: *sizek* to specify KB, *sizem* to specify MB, or *sizeg* to specify GB

Range: 10240 through 1073741824

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

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

Related Documentation	<ul style="list-style-type: none"> • Configuring Preprovisioned Member Information for a Virtual Chassis on page 44 • Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 178 • Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59
------------------------------	---

virtual-chassis (MX Series Virtual Chassis)

Syntax	<pre>virtual-chassis { member <i>member-id</i> { role (routing-engine line-card); serial-number <i>serial-number</i>; } no-split-detection; preprovisioned; traceoptions { file <i>filename</i> <files <i>number</i>> <match <i>regular-expression</i>> <no-stamp> <replace> <size maximum-file-size> <world-readable no-world-readable>; flag <i>flag</i> <detail> <disable> <receive> <send>; } }</pre>
Hierarchy Level	[edit]
Release Information	Statement introduced in Junos OS Release 11.2.
Description	<p>Create a Virtual Chassis configuration for MX Series routers.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Configuring Preprovisioned Member Information for a Virtual Chassis on page 44• Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59

PART 3

Administration

- [Administrative Tasks on page 115](#)
- [Monitoring Tasks on page 129](#)
- [Administrative Commands on page 135](#)
- [Monitoring Commands on page 145](#)

CHAPTER 8

Administrative Tasks

- [Accessing the Virtual Chassis Through the Management Interface on page 116](#)
- [Managing Files on Virtual Chassis Member Routers on page 117](#)
- [Command Forwarding in a Virtual Chassis on page 118](#)
- [Virtual Chassis Slot Number Mapping for Use with SNMP on page 126](#)

Accessing the Virtual Chassis Through the Management Interface

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

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

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



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

To access an MX Series Virtual Chassis through the management Ethernet interface, do one of the following:

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

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

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

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

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

```
{master:member0-re0}[edit groups]
```

```
user@host# set membern-ren interfaces fxp0 unit 0 family inet address
ip-address/prefix-length
```

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

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

Related Documentation

- [Configuring a Consistent Management IP Address in the Junos® OS Network Interfaces](#)

Managing Files on Virtual Chassis Member Routers

In a Virtual Chassis configuration for MX Series 3D Universal Edge Routers, you can manage files on local and remote member routers by including a member specification in the following **file** operational commands:

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

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

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

To manage files on a specific member router and a specific Routing Engine in an MX Series Virtual Chassis:

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

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

```
user@host> file option member(0 | 1)-re(0 | 1):command-option
```

For example, the following **file list** command uses the **member1-re0** specification to display a list of the files in the **/config** directory on the Routing Engine in slot 0 (**re0**) in Virtual Chassis **member 1**. The router forwards the command from **member 0**, where it is issued, to **member 1**, and displays the results as if the command were processed on the local router.

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

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

```
member1-re0:
```

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

- Related Documentation**
- [Interchassis Redundancy and Virtual Chassis Overview on page 3](#)
 - [Virtual Chassis Components Overview on page 5](#)
 - [Format for Specifying Filenames and URLs in Junos OS CLI Commands in the Junos OS System Basics Configuration Guide](#)

Command Forwarding in a Virtual Chassis

You can run some CLI commands on all member routers, on the local member router, or on a specific member router in an MX Series Virtual Chassis configuration. This feature is referred to as *command forwarding*. With command forwarding, the router sends the command to the specified member router or routers, and displays the results as if the command were processed on the local router.

For example, to collect information about your system prior to contacting Juniper Networks Technical Assistance Center (JTAC), use the command **request support information all-members** to gather data for all the member routers. If you want to gather this data only for a particular member router, use the command **request support information member member-id**.

[Table 14 on page 118](#) describes the commands that you can run on all (both) member routers (with the **all-members** option), on the local member router (with the **local** option), or on a specific member router (with the **member member-id** option) in an MX Series Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis

Command	Purpose	all-members	local	member member-id
request chassis fpc	Control the operation of the Flexible PIC Concentrator (FPC).	Change FPC status of all members of the Virtual Chassis configuration.	(Default) Change FPC status of the local Virtual Chassis member.	Change FPC status of the specified member of the Virtual Chassis configuration.
request chassis fpm resync	Resynchronize the craft interface status.	Resynchronize the craft interface status on all members of the Virtual Chassis configuration.	(Default) Resynchronize the craft interface status on the local Virtual Chassis member.	Resynchronize the craft interface status on the specified member of the Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis (*continued*)

Command	Purpose	all-members	local	member <i>member-id</i>
request chassis routing-engine master	Control which Routing Engine is the master for a router with dual Routing Engines.	Control Routing Engine mastership on the Routing Engines in all member routers of the Virtual Chassis configuration.	(Default) Control Routing Engine mastership on the Routing Engines in the local Virtual Chassis configuration.	Control Routing Engine mastership on the Routing Engines of the specified member in the Virtual Chassis configuration.
request routing-engine login	Specify a tty connection for login for a router with two Routing Engines.	Log in to all members of the Virtual Chassis configuration.	(Default) Log in to the local Virtual Chassis member.	Log in to the specified member of the Virtual Chassis configuration.
request support information	Display information about the system.	(Default) Display system information for all members of the Virtual Chassis configuration.	Display system information for the local Virtual Chassis member.	Display system information for the specified member of the Virtual Chassis configuration.
request system halt	Stop the router.	(Default) Halt all members of the Virtual Chassis configuration.	Halt the local Virtual Chassis member.	Halt the specified member of the Virtual Chassis configuration.
request system partition abort	Terminate a previously scheduled storage media partition operation.	(Default) Abort a previously scheduled storage media partition operation for all members of the Virtual Chassis configuration.	Abort a previously scheduled storage media partition operation for the local Virtual Chassis member.	Abort a previously scheduled storage media partition operation for the specified member of the Virtual Chassis member.
request system partition hard-disk	Set up the hard disk for partitioning.	(Default) Schedule a partition of the hard disk for all members of the Virtual Chassis configuration.	Schedule a partition of the hard disk for the local Virtual Chassis member.	Schedule a partition of the hard disk for the specified member of the Virtual Chassis configuration.
request system power-off	Power off the software.	(Default) Power off all members of the Virtual Chassis configuration.	Power off the local Virtual Chassis member.	Power off the specified member of the Virtual Chassis configuration.
request system reboot	Reboot the software.	(Default) Reboot the software on all members of the Virtual Chassis configuration.	Reboot the software on the local Virtual Chassis member.	Reboot the software on the specified member of the Virtual Chassis configuration.
request system snapshot	Back up the currently running and active file system partitions on the router to standby partitions that are not running.	(Default) Archive data and executable areas for all members of the Virtual Chassis configuration.	Archive data and executable areas for the local Virtual Chassis member.	Archive data and executable areas for the specified member of the Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis (*continued*)

Command	Purpose	all-members	local	member <i>member-id</i>
request system software add	Install a software package or bundle on the router.	(Default if no options specified) Install a software package on all members of the Virtual Chassis configuration.	—	Install a software package on the specified member of the Virtual Chassis configuration.
request system software rollback	Revert to the software that was loaded at the last successful request system software add command.	(Default) Attempt to roll back to the previous set of packages on all members of the Virtual Chassis configuration.	Attempt to roll back to the previous set of packages on the local Virtual Chassis member.	Attempt to roll back to the previous set of packages on the specified member of the Virtual Chassis configuration.
request system software validate	Validate candidate software against the current configuration of the router.	—	(Default if no options specified) Validate the software package on the local Virtual Chassis member.	Validate the software bundle or package on the specified member of the Virtual Chassis configuration.
request system storage cleanup	Free storage space on the router or switch by rotating log files and proposing a list of files for deletion.	(Default) Delete files on all members of the Virtual Chassis configuration.	Delete files on the local Virtual Chassis member.	Delete files on the specified member of the Virtual Chassis configuration.
restart	Restart a Junos OS process.	Restart the software process for all members of the Virtual Chassis configuration.	(Default) Restart the software process for the local Virtual Chassis member.	Restart the software process for a specified member of the Virtual Chassis configuration.
show chassis alarms	Display information about the conditions that have been configured to trigger alarms.	(Default) Display information about alarm conditions for all the member routers of the Virtual Chassis configuration.	Display information about alarm conditions for the local Virtual Chassis member.	Display information about alarm conditions for the specified member of the Virtual Chassis configuration.
show chassis craft-interface	View messages currently displayed on the craft interface.	(Default) Display information currently on the craft interface for all members of the Virtual Chassis configuration.	Display information currently on the craft interface for the specified member of the Virtual Chassis configuration.	Display information currently on the craft interface for the specified member of the Virtual Chassis configuration.
show chassis environment	Display environmental information about the router or switch chassis, including the temperature and information about the fans, power supplies, and Routing Engine.	(Default) Display chassis environmental information for all the members of the Virtual Chassis configuration.	Display chassis environmental information for the local Virtual Chassis member.	Display chassis environmental information for the specified member of the Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis (*continued*)

Command	Purpose	all-members	local	member <i>member-id</i>
show chassis environment cb	Display environmental information about the Control Boards (CBs).	(Default) Display environmental information about the CBs on all the members of the Virtual Chassis configuration.	Display environmental information about the CBs on the local Virtual Chassis member.	Display environmental information about the CBs on the specified member of the Virtual Chassis configuration.
show chassis environment fpc	Display environmental information about Flexible PIC Concentrators (FPCs).	(Default) Display environmental information for the FPCs in all the members of the Virtual Chassis configuration.	Display environmental information for the FPCs in the local Virtual Chassis member.	Display environmental information for the FPCs in the specified member of the Virtual Chassis configuration.
show chassis environment pem	Display Power Entry Module (PEM) environmental status information.	(Default) Display environmental information about the PEMs in all the member routers of the Virtual Chassis configuration.	Display environmental information about the PEMs in the local Virtual Chassis member.	Display environmental information about the PEMs in the specified member of the Virtual Chassis configuration.
show chassis environment routing-engine	Display Routing Engine environmental status information.	(Default) Display environmental information about the Routing Engines in all member routers in the Virtual Chassis configuration.	Display environmental information about the Routing Engines in the local Virtual Chassis member.	Display environmental information about the Routing Engines in the specified member of the Virtual Chassis configuration.
show chassis ethernet-switch	Display information about the ports on the Control Board (CB) Ethernet switch.	(Default) Display information about the ports on the CB Ethernet switch on all the members of the Virtual Chassis configuration.	Display information about the ports on the CB Ethernet switch on the local Virtual Chassis member.	Display information about the ports on the CB Ethernet switch on the specified member of the Virtual Chassis configuration.
show chassis fabric fpcs	Display the state of the electrical and optical switch fabric links between the Flexible PIC Concentrators (FPCs) and the Switch Interface Boards (SIBs).	(Default) Display the switching fabric link states for the FPCs in all members of the Virtual Chassis configuration.	Display the switching fabric link states for the FPCs in the local Virtual Chassis member.	Display the switching fabric link states for the FPCs in the specified member of the Virtual Chassis configuration.
show chassis fabric map	Display the switching fabric map state.	(Default) Display the switching fabric map state for all the members of the Virtual Chassis configuration.	Display the switching fabric map state for the local Virtual Chassis member.	Display the switching fabric map state for the specified member of the Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis (*continued*)

Command	Purpose	all-members	local	member <i>member-id</i>
show chassis fabric plane	Display the state of all fabric plane connections.	(Default) Display the state of all fabric plane connections on all members of the Virtual Chassis configuration.	Display the state of all fabric plane connections on the local Virtual Chassis member.	Display the state of all fabric plane connections on the specified member of the Virtual Chassis configuration.
show chassis fabric plane-location	Display the Control Board (CB) location of each plane on both the master and backup Routing Engine.	(Default) Display the CB location of each fabric plane on the Routing Engines in all member routers in the Virtual Chassis configuration.	Display the CB location of each fabric plane on the Routing Engines in the local Virtual Chassis member.	Display the CB location of each fabric plane on the Routing Engines in the specified member in the Virtual Chassis configuration.
show chassis fan	Display information about the fan tray and fans.	(Default) Display information about the fan tray and fans for all members of the Virtual Chassis configuration.	Display information about the fan tray and fans for the local Virtual Chassis member.	Display information about the fan tray and fans for the specified member of the Virtual Chassis configuration.
show chassis firmware	Display the version levels of the firmware running on the System Control Board (SCB), Switching and Forwarding Module (SFM), System and Switch Board (SSB), Forwarding Engine Board (FEB), Flexible PIC Concentrators (FPCs), and Routing Engines.	(Default) Display the version levels of the firmware running for all members of the Virtual Chassis configuration.	Display the version levels of the firmware running for the local Virtual Chassis member.	Display the version levels of the firmware running for the specified member of the Virtual Chassis configuration.
show chassis fpc	Display status information about the installed Flexible PIC Concentrators (FPCs) and PICs.	(Default) Display status information for all FPCs on all members of the Virtual Chassis configuration.	Display status information for all FPCs on the local Virtual Chassis member.	Display status information for all FPCs on the specified member of the Virtual Chassis configuration.
show chassis hardware	Display a list of all Flexible PIC Concentrators (FPCs) and PICs installed in the router or switch chassis, including the hardware version level and serial number.	(Default) Display hardware-specific information for all the members of the Virtual Chassis configuration.	Display hardware-specific information for the local Virtual Chassis member.	Display hardware-specific information for the specified member of the Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis (*continued*)

Command	Purpose	all-members	local	member <i>member-id</i>
show chassis location	Display the physical location of the chassis.	(Default) Display the physical location of the chassis for all the member routers in the Virtual Chassis configuration.	Display the physical location of the chassis for the local Virtual Chassis member.	Display the physical location of the chassis for the specified member of the Virtual Chassis configuration.
show chassis mac-addresses	Display the media access control (MAC) addresses for the router, switch chassis, or switch.	(Default) Display the MAC addresses for all the member routers of the Virtual Chassis configuration.	Display the MAC addresses for the local Virtual Chassis member.	Display the MAC addresses for the specified member of the Virtual Chassis configuration.
show chassis pic	Display status information about the PIC installed in the specified Flexible PIC Concentrator (FPC) and PIC slot.	(Default) Display PIC information for all member routers in the Virtual Chassis configuration.	Display PIC information for the local Virtual Chassis member.	Display PIC information for the specified member of the Virtual Chassis configuration.
show chassis power	Display power limits and usage information for the AC or DC power sources.	(Default) Display power usage information for all members of the Virtual Chassis configuration.	Display power usage information for the local Virtual Chassis member.	Display power usage information for the specified member of the Virtual Chassis configuration.
show chassis routing-engine	Display the status of the Routing Engine.	(Default) Display Routing Engine information for all members of the Virtual Chassis configuration.	Display Routing Engine information for the local Virtual Chassis member.	Display Routing Engine information for the specified member of the Virtual Chassis configuration.
show chassis temperature-thresholds	Display chassis temperature threshold settings, in degrees Celsius.	(Default) Display the chassis temperature threshold settings of all member routers in the Virtual Chassis configuration.	Display the chassis temperature threshold settings of the local Virtual Chassis member.	Display the chassis temperature threshold settings of the specified member of the Virtual Chassis configuration.
show pfe fpc	Display Packet Forwarding Engine statistics for the specified Flexible PIC Concentrator (FPC).	(Default) Display Packet Forwarding Engine statistics for the specified FPC in all members of the Virtual Chassis configuration.	Display Packet Forwarding Engine statistics for the specified FPC in the local Virtual Chassis member.	Display Packet Forwarding Engine statistics for the specified FPC in the specified member of the Virtual Chassis configuration.
show pfe terse	Display Packet Forwarding Engine status information.	(Default) Display Packet Forwarding Engine status information for all members in the Virtual Chassis configuration.	Display Packet Forwarding Engine status information for the local Virtual Chassis member.	Display Packet Forwarding Engine status information for the specified member of the Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis (*continued*)

Command	Purpose	all-members	local	member <i>member-id</i>
show system audit	Display the state and checksum values for file systems.	(Default) Display file system MD5 hash and permissions information on all members of the Virtual Chassis configuration.	Display file system MD5 hash and permissions information on the local Virtual Chassis member.	Display file system MD5 hash and permissions information on the specified member of the Virtual Chassis configuration.
show system boot-messages	Display initial messages generated by the system kernel upon startup.	(Default) Display boot time messages on all members of the Virtual Chassis configuration.	Display boot time messages on the local Virtual Chassis member.	Display boot time messages on the specified member of the Virtual Chassis configuration.
show system buffers	Display information about the buffer pool that the Routing Engine uses for local traffic.	(Default) Show buffer statistics for all members of the Virtual Chassis configuration.	Show buffer statistics for the local Virtual Chassis member.	Show buffer statistics for the specified member of the Virtual Chassis configuration.
show system connections	Display information about the active IP sockets on the Routing Engine.	(Default) Display system connection activity for all members of the Virtual Chassis configuration.	Display system connection activity for the local Virtual Chassis member.	Display system connection activity for the specified member of the Virtual Chassis configuration.
show system directory-usage	Display directory usage information.	Display directory information for all members of the Virtual Chassis configuration.	(Default) Display directory information for the local Virtual Chassis member.	Display directory information for the specified member of the Virtual Chassis configuration.
show system processes	Display information about software processes that are running on the router and that have controlling terminals.	(Default) Display standard system process information for all members of the Virtual Chassis configuration.	Display standard system process information for the local Virtual Chassis member.	Display standard system process information for the specified member of the Virtual Chassis configuration.
show system queues	Display queue statistics.	(Default) Display system queue statistics for all members of the Virtual Chassis configuration.	Display system queue statistics for the local Virtual Chassis member.	Display system queue statistics for the specified member of the Virtual Chassis configuration.
show system reboot	Display pending system reboots or halts.	(Default) Display halt or reboot request information for all members of the Virtual Chassis configuration.	Display halt or reboot request information for the local Virtual Chassis member.	Display halt or reboot request information for the specified member of the Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis (*continued*)

Command	Purpose	all-members	local	member <i>member-id</i>
show system statistics	Display system-wide protocol-related statistics.	(Default) Display system statistics for a protocol for all members of the Virtual Chassis configuration.	Display system statistics for a protocol for the local Virtual Chassis member.	Display system statistics for a protocol for the specified member of the Virtual Chassis configuration.
show system storage	Display statistics about the amount of free disk space in the router's file systems.	(Default) Display system storage statistics for all members of the Virtual Chassis configuration.	Display system storage statistics for the local Virtual Chassis member.	Display system storage statistics for the specified member of the Virtual Chassis configuration.
show system switchover	Display whether graceful Routing Engine switchover is configured, the state of the kernel replication (ready or synchronizing), any replication errors, and whether the primary and standby Routing Engines are using compatible versions of the kernel database.	(Default) Display graceful Routing Engine switchover information for all Routing Engines on all members of the Virtual Chassis configuration.	Display graceful Routing Engines switchover information for all Routing Engines on the local Virtual Chassis member.	Display graceful Routing Engine switchover information for all Routing Engines on the specified member of the Virtual Chassis configuration.
show system uptime	Display the current time and information about how long the router or switch, router or switch software, and routing protocols have been running.	(Default) Show time since the system rebooted and processes started on all members of the Virtual Chassis configuration.	Show time since the system rebooted and processes started on the local Virtual Chassis member.	Show time since the system rebooted and processes started on the specified member of the Virtual Chassis configuration.
show system users	List information about the users who are currently logged in to the router.	(Default) Display users currently logged in to all members of the Virtual Chassis configuration.	Display users currently logged in to the local Virtual Chassis member.	Display users currently logged in to the specified member of the Virtual Chassis configuration.
show system virtual-memory	Display the usage of Junos OS kernel memory listed first by size of allocation and then by type of usage.	(Default) Display kernel dynamic memory usage information for all members of the Virtual Chassis configuration.	Display kernel dynamic memory usage information for the local Virtual Chassis member.	Display kernel dynamic memory usage information for the specified member of the Virtual Chassis configuration.
show version	Display the hostname and version information about the software running on the router.	(Default) Display standard information about the hostname and version of the software running on all members of the Virtual Chassis configuration.	Display standard information about the hostname and version of the software running on the local Virtual Chassis member.	Display standard information about the hostname and version of the software running on the specified member of the Virtual Chassis configuration.

Table 14: Commands Available for Command Forwarding in an MX Series Virtual Chassis (*continued*)

Command	Purpose	all-members	local	member <i>member-id</i>
show version invoke-on	Display the hostname and version information about the software running on a router with two Routing Engines.	(Default) Display the hostname and version information about the software running on all master and backup Routing Engines on all members of the Virtual Chassis configuration.	Display the hostname and version information about the software running on all master and backup Routing Engines on the local Virtual Chassis member.	Display the hostname and version information about the software running on all master and backup Routing Engines on the specified member of the Virtual Chassis configuration.

- Related Documentation**
- [Virtual Chassis Components Overview on page 5](#)
 - [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
 - Junos OS Operational Mode Commands

Virtual Chassis Slot Number Mapping for Use with SNMP

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

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

[Table 15 on page 126](#) lists the jnxFruSlot number that appears in the jnxFruTable of the jnxBoxAnatomy MIB, and the corresponding line card physical slot number in each member router of a two-member MX Series Virtual Chassis. For example, a jnxFruSlot value of 15 corresponds to physical slot 3 in member 0 of an MX Series Virtual Chassis. A jnxFruSlot value of 30 corresponds to physical slot 6 in member 1 of an MX Series Virtual Chassis.

Table 15: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series Virtual Chassis

jnxFruSlot Number	Line Card Slot Number	MX Series Virtual Chassis Member ID
Line Cards in MX Series Virtual Chassis Member ID 0 (offset = 12):		
12	0	0
13	1	0
14	2	0

Table 15: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series Virtual Chassis (*continued*)

jnxFruSlot Number	Line Card Slot Number	MX Series Virtual Chassis Member ID
15	3	0
16	4	0
17	5	0
18	6	0
19	7	0
20	8	0
21	9	0
22	10	0
23	11	0
Line Cards in MX Series Virtual Chassis Member ID 1 (offset = 24)		
24	0	1
25	1	1
26	2	1
27	3	1
28	4	1
29	5	1
30	6	1
31	7	1
32	8	1
33	9	1
34	10	1
35	11	1

- Related Documentation**
- [Virtual Chassis Components Overview on page 5](#)
 - [SNMP MIBs and Traps Reference](#)

CHAPTER 9

Monitoring Tasks

- [Verifying the Status of Virtual Chassis Member Routers on page 129](#)
- [Verifying the Operation of Virtual Chassis Ports on page 129](#)
- [Verifying Neighbor Reachability for Member Routers in a Virtual Chassis on page 130](#)
- [Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis on page 130](#)
- [Viewing Information in the Virtual Chassis Control Protocol Adjacency Database on page 131](#)
- [Viewing Information in the Virtual Chassis Control Protocol Link-State Database on page 131](#)
- [Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database on page 132](#)
- [Viewing Virtual Chassis Control Protocol Routing Tables on page 132](#)
- [Viewing Virtual Chassis Control Protocol Statistics for Member Routers and Virtual Chassis Ports on page 133](#)

Verifying the Status of Virtual Chassis Member Routers

Purpose	Verify that the member routers in an MX Series Virtual Chassis are properly configured.
Action	Display the status of the members of the Virtual Chassis configuration: <code>user@host> show virtual-chassis status</code>
Related Documentation	<ul style="list-style-type: none">• Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33• Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59

Verifying the Operation of Virtual Chassis Ports

Purpose	Verify that the Virtual Chassis ports in an MX Series Virtual Chassis are properly configured and operational.
Action	<ul style="list-style-type: none">• To display the status of the Virtual Chassis ports for both member routers in the Virtual Chassis:

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

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

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

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

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

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Verifying Neighbor Reachability for Member Routers in a Virtual Chassis

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

Action • To display neighbor reachability information for both member routers in the Virtual Chassis:

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

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

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

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

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

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis

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

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

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

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

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

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

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

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Viewing Information in the Virtual Chassis Control Protocol Adjacency Database

Purpose View information about neighbors in the Virtual Chassis Control Protocol (VCCP) adjacency database for an MX Series Virtual Chassis configuration.

Action • To display VCCP neighbor adjacency information for both member routers in the Virtual Chassis:

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

- To display VCCP neighbor adjacency information for a specified member router in the Virtual Chassis:

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

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

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

- To display VCCP neighbor adjacency information for the device with a specified system ID on the specified member router:

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

- To display VCCP neighbor adjacency information for the member router on which you are issuing the command:

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

Related Documentation

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

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

Purpose View information about protocol data unit (PDU) packets in the Virtual Chassis Control Protocol (VCCP) link-state database for an MX Series Virtual Chassis configuration.

Action • To display VCCP PDU information for both member routers in the Virtual Chassis:

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

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

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

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

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

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

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

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

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

**Related
Documentation**

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

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

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

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

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

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

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

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

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

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

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

**Related
Documentation**

- [Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Viewing Virtual Chassis Control Protocol Routing Tables

Purpose View Virtual Chassis Control Protocol (VCCP) unicast and multicast routing tables for an MX Series Virtual Chassis configuration.

Action	<ul style="list-style-type: none"> To display the VCCP unicast and multicast routing tables for both member routers in the Virtual Chassis: <pre>user@host> show virtual-chassis protocol route all-members</pre> To display the VCCP unicast and multicast routing tables for a specified member router in the Virtual Chassis: <pre>user@host> show virtual-chassis protocol route member member-id</pre> To display the VCCP unicast and multicast routing tables to the destination with the specified system ID: <pre>user@host> show virtual-chassis protocol route destination-id</pre> To display the VCCP unicast and multicast routing tables to the destination with the specified system ID on the specified member router: <pre>user@host> show virtual-chassis protocol route member member-id destination-id</pre> To display the VCCP unicast and multicast routing tables for the member router on which you are issuing the command: <pre>user@host> show virtual-chassis protocol route local</pre>
Related Documentation	<ul style="list-style-type: none"> Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33 Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59

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

Purpose	View Virtual Chassis Control Protocol (VCCP) statistics for one or both member routers, or for a specified Virtual Chassis port interface, in an MX Series Virtual Chassis configuration.
Action	<ul style="list-style-type: none"> To display VCCP statistics for both member routers in the Virtual Chassis: <pre>user@host> show virtual-chassis protocol statistics all-members</pre> To display VCCP statistics for a specified member router in the Virtual Chassis: <pre>user@host> show virtual-chassis protocol statistics member member-id</pre> To display VCCP statistics for a specified Virtual Chassis port interface: <pre>user@host> show virtual-chassis protocol statistics vcp-slot/pic/port.logical-unit-number</pre> To display VCCP statistics for the member router on which you are issuing the command: <pre>user@host> show virtual-chassis protocol statistics local</pre>
Related Documentation	<ul style="list-style-type: none"> Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33 Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59

CHAPTER 10

Administrative Commands

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

Syntax request virtual-chassis member-id delete

Release Information Command introduced in Junos OS Release 11.2.

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



.....

NOTE: Issuing the command to remove the member ID causes the router to reboot, and requires you to confirm that you want to proceed with this operation. If you do not confirm the operation, the software cancels the command.

.....

Required Privilege Level system-control

Related Documentation


- [Deleting Member IDs in a Virtual Chassis Configuration on page 52](#)
- [Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 73](#)

List of Sample Output [request virtual-chassis member-id delete on page 136](#)

Sample Output

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



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

Syntax	<code>request virtual-chassis member-id set member <i>member-id</i></code>
Release Information	Command introduced in Junos OS Release 11.2.
Description	Assign (set) a member ID to an MX Series router that you want to add as a member of a Virtual Chassis configuration.
	 <p>NOTE: Issuing the command to assign a member ID causes the router to reboot, and requires you to confirm that you want to proceed with this operation. If you do not confirm the operation, the software cancels the command.</p>
Options	member <i>member-id</i> —Numeric value that identifies a member router in a Virtual Chassis configuration. When you assign a member ID to a router, assign the same member ID defined for this router in the MX Series Virtual Chassis preprovisioned configuration. Replace <i>member-id</i> with the value 0 or 1.
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none"> • Configuring Member IDs for a Virtual Chassis on page 47 • Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59
List of Sample Output	request virtual-chassis member-id set on page 137

Sample Output

```
request virtual-chassis member-id set
user@host> request virtual-chassis member-id set member 0
This command will enable virtual-chassis mode and reboot the system.
Continue? [yes,no] (no)
```

request virtual-chassis routing-engine master switch (MX Series Virtual Chassis)

Syntax	request virtual-chassis routing-engine master switch <check>
Release Information	Command introduced in Junos OS Release 11.2. Option check introduced in Junos OS Release 12.2.
Description	<p>Change the mastership in an MX Series Virtual Chassis by switching the global roles of the master router and backup router in the Virtual Chassis configuration. The request virtual-chassis routing-engine master switch command must be issued from the master router (VC-Mm).</p> <p>For MX Series routers with dual Routing Engines in a Virtual Chassis, the local roles (master and standby) of the Routing Engines in each member router do not change after you issue the request virtual-chassis routing-engine master switch command.</p> <div><p>NOTE: Before you issue the request virtual-chassis routing-engine master switch command from the master router in the Virtual Chassis, make sure that the system configuration is synchronized between the master router and backup router. If the configuration on the master router and backup router is not synchronized, or if you attempt to issue the request virtual-chassis routing-engine master switch command from the backup router instead of from the master router, the router displays an error message and rejects the command.</p><p>If you issue the request virtual-chassis routing-engine master switch command when the Virtual Chassis is in a transition state (for example, the backup router is disconnecting from the Virtual Chassis), the router does not process the command.</p></div>
Options	check —(Optional) Perform a check from the master router in an MX Series Virtual Chassis to determine whether the member routers are ready for GRES from a database synchronization perspective, without initiating the GRES operation itself.
Required Privilege Level	system-control
Related Documentation	<ul style="list-style-type: none">• Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 54• Determining GRES Readiness in a Virtual Chassis Configuration on page 55• Mastership Election in a Virtual Chassis on page 15
List of Sample Output	request virtual-chassis routing-engine master switch (From Master Router) on page 139

[request virtual-chassis routing-engine master switch \(Error When Configuration Not Synchronized\) on page 139](#)

[request virtual-chassis routing-engine master switch \(Error When Run from Backup Router\) on page 139](#)

[request virtual-chassis routing-engine master switch check \(Ready for GRES\) on page 139](#)

[request virtual-chassis routing-engine master switch check \(Not Ready for GRES\) on page 139](#)

Sample Output

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

```
{master:member0-re0}
user@host> request virtual-chassis routing-engine master switch
Do you want to continue ? [yes,no] (no)
```

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

```
{master:member0-re0}
user@host> request virtual-chassis routing-engine master switch
Error: mastership switch request NOT honored, backup not ready
```

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

```
{backup:member1-re0}
user@host1> request virtual-chassis routing-engine master switch
error: Virtual Chassis member is not the protocol master
```

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

```
{master:member0-re0}
user@host> request virtual-chassis routing-engine master switch check
{master:member0-re0}
```

[request virtual-chassis routing-engine master switch check \(Not Ready for GRES\)](#)

```
{master:member0-re0}
user@host> request virtual-chassis routing-engine master switch check
error: chassisd Not ready for mastership switch, try after 217 secs.
mastership switch request NOT honored, backup not ready
```

request virtual-chassis vc-port delete (MX Series Virtual Chassis)

Syntax request virtual-chassis vc-port delete fpc-slot *fpc-slot-number* pic-slot *pic-slot-number* port *port-number*
<(local | member *member-id*)>

Release Information Command introduced in Junos OS Release 11.2.

Description Remove (**delete**) a Virtual Chassis port from a member router in an MX Series Virtual Chassis configuration. After a Virtual Chassis port is created, it is renamed **vcp-slot/pic/port**, and is no longer available for configuration as a standard network port. After you remove a Virtual Chassis port, it becomes available to the global configuration and can again function as a standard network port.



NOTE: If the member ID has not been set on the router where you issue the **request virtual-chassis vc-port delete** command, the software prevents the removal of the Virtual Chassis port on the router. To set the member ID, use the **request virtual-chassis member-id set** command.

Options **fpc-slot** *fpc-slot-number*—Number of the Flexible PIC Concentrator (FPC) slot on which the Virtual Chassis port resides. The slot number corresponds to the Modular Port Concentrator (MPC) slot number. Replace *fpc-slot-number* with a value appropriate for your router:

- MX960 router—0 through 11.
- MX480 router—0 through 5.
- MX240 router—0 through 2.

pic-slot *pic-slot-number*—Number of the PIC slot on which the Virtual Chassis port resides. Replace *pic-slot-number* with a value in the range 0 through 3.

port *port-number*—Number of the port on the PIC on which the Virtual Chassis port resides. Replace *port-number* with a value appropriate for your PIC.

local—(Optional) Delete the Virtual Chassis port on the member router on which you are issuing the command. This is the default behavior if you do not specify the **local** or **member** options.

member *member-id*—(Optional) Numeric value that identifies the remote Virtual Chassis member on which you want to delete the Virtual Chassis port. Replace *member-id* with the value 0 or 1.

Required Privilege Level system-control

- Related Documentation**
- [Deleting Virtual Chassis Ports in a Virtual Chassis Configuration on page 51](#)
 - [Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 73](#)

List of Sample Output [request virtual-chassis vc-port delete \(Remove vcp-3/2/1\) on page 141](#)

Sample Output

[request virtual-chassis vc-port delete \(Remove vcp-3/2/1\)](#)

```
user@host> request virtual-chassis vc-port delete fpc-slot 3 pic-slot 2 port 1
vc-port successfully deleted
```

request virtual-chassis vc-port set (MX Series Virtual Chassis)

Syntax request virtual-chassis vc-port set fpc-slot *fpc-slot-number* pic-slot *pic-slot-number* port *port-number*
 <(local | member *member-id*)>

Release Information Command introduced in Junos OS Release 11.2.

Description Create (**set**) a Virtual Chassis port on an MX Series router through which the router connects to other member routers in the Virtual Chassis. You can create Virtual Chassis ports only on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) network ports on MX Series routers.

After a Virtual Chassis port is created, it is renamed **vcp-slot/pic/port**, and is no longer available for configuration as a standard network port. Virtual Chassis ports can be used only to interconnect the MX Series routers in the Virtual Chassis.



NOTE: If the member ID has not been set on the router where you issue the **request virtual-chassis vc-port set** command, the software prevents the creation of the Virtual Chassis port on the router. To set the member ID, use the **request virtual-chassis member-id set** command.

Options **fpc-slot *fpc-slot-number***—Number of the Flexible PIC Concentrator (FPC) slot on which the Virtual Chassis port resides. The slot number corresponds to the Modular Port Concentrator (MPC) slot number. Replace ***fpc-slot-number*** with a value appropriate for your router:

- MX960 router—0 through 11.
- MX480 router—0 through 5.
- MX240 router—0 through 2.

When you issue the **show interfaces** command on a member router in an MX Series Virtual Chassis, the FPC slot number displayed in the command output reflects the FPC slot numbering and offset used in the Virtual Chassis instead of the physical slot number where the FPC is actually installed. The router with member ID 0 in the Virtual Chassis uses FPC slot numbers 0 through 11 with no offset, and the router with member ID 1 uses FPC slot numbers 12 through 23, with an offset of 12. For example, a 10-Gigabit Ethernet interface that appears as **xe-14/2/2** (FPC slot 14, PIC slot 2, port 2) in the **show interfaces** command is actually interface **xe-2/2/2** (FPC slot 2, PIC slot 2, port 2) on member ID 1 after deducting the FPC slot numbering offset of 12 for member ID 1.

pic-slot *pic-slot-number*—Number of the PIC slot on which the Virtual Chassis port resides. Replace ***pic-slot-number*** with a value in the range 0 through 3.

port *port-number*—Number of the port on the PIC on which the Virtual Chassis port resides. Replace ***port-number*** with a value appropriate for your PIC.

local—(Optional) Set the Virtual Chassis port on the member router on which you are issuing the command. This is the default behavior if you do not specify the **local** or **member** options.

member *member-id*—(Optional) Numeric value that identifies the remote Virtual Chassis member on which you want to create the Virtual Chassis port. Replace ***member-id*** with the value 0 or 1.

Required Privilege Level system-control

Related Documentation

- [Configuring Virtual Chassis Ports to Interconnect Member Routers on page 49](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)
- [Guidelines for Configuring Virtual Chassis Ports on page 10](#)

List of Sample Output

[request virtual-chassis vc-port set \(No Existing Network Port\) on page 143](#)
[request virtual-chassis vc-port set \(Existing Network Port Converted\) on page 143](#)
[request virtual-chassis vc-port set \(On Local Router\) on page 143](#)
[request virtual-chassis vc-port set \(On Remote Member Router 1\) on page 143](#)

Sample Output

[request virtual-chassis vc-port set \(No Existing Network Port\)](#)

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

[request virtual-chassis vc-port set \(Existing Network Port Converted\)](#)

```
user@host> request virtual-chassis vc-port set fpc-slot 2 pic-slot 1 port 1
Port conversion initiated, use "show virtual chassis vc-port" to verify
```

[request virtual-chassis vc-port set \(On Local Router\)](#)

```
user@host> request virtual-chassis vc-port set fpc-slot 2 pic-slot 1 port 3 local
vc-port successfully set
```

[request virtual-chassis vc-port set \(On Remote Member Router 1\)](#)

```
user@host> request virtual-chassis vc-port set fpc-slot 5 pic-slot 3 port 10 member 1
vc-port successfully set
```


CHAPTER 11

Monitoring Commands

show virtual-chassis active-topology (MX Series Virtual Chassis)

Syntax	show virtual-chassis active-topology <(all-members local member <i>member-id</i>)>
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display information about neighbor reachability from each member router in an MX Series Virtual Chassis configuration. You can issue the show virtual-chassis active-topology command from the console of either member router in the Virtual Chassis.
Options	<p>all-members—(Optional) Display neighbor reachability information for both member routers in a Virtual Chassis configuration. This is the default behavior if no options are specified.</p> <p>local—(Optional) Display neighbor reachability information for the member router on which you are issuing the command.</p> <p>member <i>member-id</i>—(Optional) Display neighbor reachability information for the specified Virtual Chassis member router. Replace <i>member-id</i> with the value 0 or 1.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Verifying Neighbor Reachability for Member Routers in a Virtual Chassis on page 130
List of Sample Output	show virtual-chassis active-topology all-members on page 147 show virtual-chassis active-topology local on page 147 show virtual-chassis active-topology member 1 on page 147
Output Fields	Table 16 on page 146 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 16: show virtual-chassis active-topology Output Fields

Field Name	Field Description
membern	Member ID of the Virtual Chassis member router for which output is displayed.
Destination ID	Member ID of the destination (neighbor) router.
Next-hop	Member ID and Virtual Chassis port interface (in the format vcp-slot/pic/port.logical-unit-number) of the next-hop to which the router forwards packets for the destination ID.

Sample Output

show virtual-chassis
active-topology
all-members

```
{master:member0-re0}
user@host> show virtual-chassis active-topology all-members
member0:
```

Destination ID	Next-hop
1	1(vcp-5/0/0.32768)

```
member1:
```

Destination ID	Next-hop
0	0(vcp-1/3/0.32768)

show virtual-chassis
active-topology local

```
{master:member0-re0}
user@host> show virtual-chassis active-topology local
```

Destination ID	Next-hop
1	1(vcp-5/0/0.32768)

show virtual-chassis
active-topology
member 1

```
{master:member0-re0}
user@host> show virtual-chassis active-topology member 1
member1:
```

Destination ID	Next-hop
0	0(vcp-1/3/0.32768)

show virtual-chassis device-topology (MX Series Virtual Chassis)

Syntax	<code>show virtual-chassis device-topology</code> <code><(all-members local member <i>member-id</i>)></code>
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display information about neighbor reachability for each hardware device in an MX Series Virtual Chassis configuration. On the MX Series router, there is only one active device for each member router. You can issue the show virtual-chassis device-topology command from the console of either member router in the Virtual Chassis.
Options	<p>all-members—(Optional) Display neighbor reachability information for the device in both member routers in a Virtual Chassis configuration.</p> <p>local—(Optional) Display neighbor reachability information for the device in the member router on which you are issuing the command. This is the default behavior if no options are specified.</p> <p>member <i>member-id</i>—(Optional) Display neighbor reachability information for the device in the specified Virtual Chassis member router. Replace <i>member-id</i> with the value 0 or 1.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis on page 130
List of Sample Output	show virtual-chassis device-topology all-members on page 149 show virtual-chassis device-topology local on page 149 show virtual-chassis device-topology member 1 on page 149
Output Fields	Table 17 on page 148 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 17: show virtual-chassis device-topology Output Fields

Field Name	Field Description
membern	Member ID of the Virtual Chassis member router for which output is displayed.
Member	Identifier assigned to the member router in the preprovisioned Virtual Chassis configuration.
Device	<p>Identifier assigned to the device in the member router.</p> <p>Because there is only one active device per member router in an MX Series Virtual Chassis configuration, the values in the Device and Member fields are identical.</p>

Table 17: show virtual-chassis device-topology Output Fields (*continued*)

Field Name	Field Description
Status	Status of the device: <ul style="list-style-type: none"> • Prsnt—Device is currently connected to the Virtual Chassis. • NotPrsnt—Device is not currently connected to the Virtual Chassis.
System ID	Unique identifier derived from the device's media access control (MAC) address. The System ID is included in each Virtual Chassis Control Protocol (VCCP) packet to identify the packet owner to all members of the Virtual Chassis.
Neighbor List Member/Device/Interface	Member IDs, Device IDs, and Virtual Chassis port interfaces (in the format vcp-slot/pic/port) to which this device is connected.

Sample Output

show virtual-chassis
device-topology
all-members

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

```
user@host> show virtual-chassis device-topology all-members
member0:
```

```
-----
Member  Device  Status  System ID      Neighbor List
                                Member  Device  Interface
    0      0    Prsnt   b0c6.9abf.6800    1      1    vcp-5/0/0
    1      1    Prsnt   001d.b510.0800    0      0    vcp-1/3/0
```

```
member1:
```

```
-----
Member  Device  Status  System ID      Neighbor List
                                Member  Device  Interface
    0      0    Prsnt   b0c6.9abf.6800    1      1    vcp-5/0/0
    1      1    Prsnt   001d.b510.0800    0      0    vcp-1/3/0
```

show virtual-chassis
device-topology local

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

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

```
-----
Member  Device  Status  System ID      Neighbor List
                                Member  Device  Interface
    0      0    Prsnt   b0c6.9abf.6800    1      1    vcp-5/0/0
    1      1    Prsnt   001d.b510.0800    0      0    vcp-1/3/0
```

show virtual-chassis
device-topology
member 1

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

```
user@host> show virtual-chassis device-topology member 1
member1:
```

```
-----
Member  Device  Status  System ID      Neighbor List
                                Member  Device  Interface
    0      0    Prsnt   b0c6.9abf.6800    1      1    vcp-5/0/0
    1      1    Prsnt   001d.b510.0800    0      0    vcp-1/3/0
```

show virtual-chassis protocol adjacency (MX Series Virtual Chassis)

Syntax	show virtual-chassis protocol adjacency <(brief detail extensive)> <(all-members local member <i>member-id</i>)> < <i>system-id</i> >
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display the entries (neighbors) in the Virtual Chassis Control Protocol (VCCP) adjacency database for an MX Series Virtual Chassis configuration. You can issue the show virtual-chassis protocol adjacency command from the console of either member router in the Virtual Chassis.
Options	<p>brief detail extensive—(Optional) Display the specified level of output. Using the brief option is equivalent to issuing the command with no options (the default). The detail option provides more output than the brief option. The extensive option provides complete output and is most useful for customer support personnel.</p> <p>all-members—(Optional) Display the VCCP adjacency database for both member routers in a Virtual Chassis. This is the default behavior if no options are specified.</p> <p>local—(Optional) Display the VCCP adjacency database for the member router on which you are issuing the command.</p> <p>member <i>member-id</i>—(Optional) Display the VCCP adjacency database for the specified member router. Replace <i>member-id</i> with the value 0 or 1.</p> <p><i>system-id</i>—(Optional) Display the VCCP adjacency database for the device with the specified system ID.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> Viewing Information in the Virtual Chassis Control Protocol Adjacency Database on page 131
List of Sample Output	show virtual-chassis protocol adjacency all-members brief on page 153 show virtual-chassis protocol adjacency member 0 detail on page 153 show virtual-chassis protocol adjacency member 0 detail 001d.b510.0800 on page 153 show virtual-chassis protocol adjacency local extensive on page 153
Output Fields	Table 18 on page 150 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 18: show virtual-chassis protocol adjacency Output Fields

Field Name	Field Description	Level of Output
membern	Member ID of the Virtual Chassis member router for which output is displayed.	All levels

Table 18: show virtual-chassis protocol adjacency Output Fields (*continued*)

Field Name	Field Description	Level of Output
Interface	Name of the Virtual Chassis port interface, in the format <i>vcp-slot/pic/port.logical-unit-number</i> .	brief
System	System ID of the device associated with the Virtual Chassis port interface. The System ID is derived from the device's media access control (MAC) address.	brief
State	State of the adjacency: <ul style="list-style-type: none"> • Up—The adjacency is up. • Down—The adjacency is down. 	All levels
Hold (secs)	Remaining hold time of the adjacency, in seconds.	brief
system-id	System ID of the device associated with the Virtual Chassis port interface. The System ID is derived from the device's media access control (MAC) address.	detail, extensive
interface-name	Name of the Virtual Chassis port interface, in the format <i>vcp-slot/pic/port.logical-unit-number</i> .	detail, extensive
Expires in	Number of seconds until the adjacency expires.	detail, extensive
Priority	Priority to become the designated intermediate system.	detail, extensive
Up/Down transitions	Count of adjacency status changes from Up to Down or from Down to Up .	detail, extensive
Last transition	Time of the last Up/Down transition.	detail, extensive

Table 18: show virtual-chassis protocol adjacency Output Fields (*continued*)

Field Name	Field Description	Level of Output
Transition log	<p>List of recent adjacency transitions, including:</p> <ul style="list-style-type: none"> • When—Date and time at which a VCCP adjacency transition occurred. • State—Current state of the VCCP adjacency: <ul style="list-style-type: none"> • Up—Adjacency is up and operational. • Down—Adjacency is down and not available. • Rejected—Adjacency has been rejected. • Event—Type of transition that occurred: <ul style="list-style-type: none"> • Seenself—Possible routing loop has been detected. • Interface down—Virtual Chassis port interface has gone down and is no longer available. • Error—Adjacency error. • Down reason—Reason that a VCCP adjacency is down: <ul style="list-style-type: none"> • 3-Way Handshake Failed—Connection establishment failed. • Address Mismatch—Address mismatch caused link failure. • Aged Out—Link expired. • ISO Area Mismatch—VCCP area mismatch caused link failure. • Bad Hello—Unacceptable hello message caused link failure. • BFD Session Down—Bidirectional failure detection caused link failure. • Interface Disabled—Virtual Chassis port interface is disabled. • Interface Down—Virtual Chassis port interface is unavailable. • Interface Level Disabled—VCCP level is disabled. • Level Changed—VCCP level has changed on the adjacency. • Level Mismatch—Levels on adjacency are not compatible. • MPLS LSP Down—Label-switched path (LSP) is unavailable. • MT Topology Changed—VCCP topology has changed. • MT Topology Mismatch—VCCP topology is mismatched. • Remote System ID Changed—Adjacency peer system ID changed. • Protocol Shutdown—VCCP is disabled. • CLI Command—Adjacency brought down by user. • Unknown—Unknown. 	extensive

Sample Output

show virtual-chassis
protocol adjacency
all-members brief

```
{master:member0-re0}
user@host> show virtual-chassis protocol adjacency all-members brief
member0:
-----
Interface          System          State          Hold (secs)
vcp-5/0/0.32768    001d.b510.0800 Up              57

member1:
-----
Interface          System          State          Hold (secs)
vcp-1/3/0.32768    b0c6.9abf.6800 Up              58
```

show virtual-chassis
protocol adjacency
member 0 detail

```
{master:member0-re0}
user@host> show virtual-chassis protocol adjacency member 0 detail
member0:
-----

001d.b510.0800
  interface-name: vcp-5/0/0.32768, State: Up, Expires in 57 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 18:50:41 ago
```

show virtual-chassis
protocol adjacency
member 0 detail
001d.b510.0800

```
{master:member0-re0}
user@host> show virtual-chassis protocol adjacency member 0 detail 001d.b510.0800
member0:
-----

001d.b510.0800
  interface-name: vcp-5/0/0.32768, State: Up, Expires in 58 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 18:52:08 ago
```

show virtual-chassis
protocol adjacency
local extensive

```
{master:member0-re0}
user@host> show virtual-chassis protocol adjacency local extensive

001d.b510.0800
  interface-name: vcp-5/0/0.32768, State: Up, Expires in 59 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 18:52:40 ago
  Transition log:
  When                State    Event          Down reason
  Mon Sep 20 17:26:44  Up      Seenself
```

show virtual-chassis protocol database (MX Series Virtual Chassis)

Syntax	show virtual-chassis protocol database <(brief detail extensive)> <(all-members local member <i>member-id</i>)> < <i>system-id</i> >
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display the entries in the Virtual Chassis Control Protocol (VCCP) link-state database for an MX Series Virtual Chassis configuration. The VCCP link-state database contains information about protocol data unit (PDU) packets. You can issue the show virtual-chassis protocol database command from the console of either member router in the Virtual Chassis.
Options	<p>brief detail extensive—(Optional) Display the specified level of output. Using the brief option is equivalent to issuing the command with no options (the default). The detail option provides more output than the brief option. The extensive option provides complete output and is most useful for customer support personnel.</p> <p>all-members—(Optional) Display the VCCP link-state database for both member routers in a Virtual Chassis. This is the default behavior if no options are specified.</p> <p>local—(Optional) Display the VCCP link-state database for the member router on which you are issuing the command.</p> <p>member <i>member-id</i>—(Optional) Display the VCCP link-state database for the specified member router. Replace <i>member-id</i> with the value 0 or 1.</p> <p><i>system-id</i>—(Optional) Display the VCCP link-state database for the neighbor with the specified system ID.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> Viewing Information in the Virtual Chassis Control Protocol Link-State Database on page 131
List of Sample Output	show virtual-chassis protocol database all-members brief on page 157 show virtual-chassis protocol database member 0 detail on page 157 show virtual-chassis protocol database member 0 b0c6.9abf.6800 detail on page 157 show virtual-chassis protocol database member 0 extensive on page 158
Output Fields	Table 19 on page 154 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 19: show virtual-chassis protocol database Output Fields

Field Name	Field Description	Level of Output
membern	Member ID of the Virtual Chassis member router for which output is displayed.	All levels

Table 19: show virtual-chassis protocol database Output Fields (*continued*)

Field Name	Field Description	Level of Output
LSP ID	Link-state PDU (LSP) identifier.	All levels
Sequence	Sequence number of the link-state PDU.	All levels
Checksum	Checksum value of the link-state PDU.	All levels
Lifetime	Remaining lifetime of the link-state PDU, in seconds.	All levels
number LSPs	Total number of link-state PDUs in the specified link-state database.	none, brief
Neighbor, Neighbor Info	Media access control (MAC) address of the neighbor on the advertising system.	detail, extensive
Interface	Name of the Virtual Chassis port interface, in the format <i>vcp-slot/pic/port.logical-unit-number</i> .	detail, extensive
Metric	Metric value of the prefix or neighbor.	detail, extensive
Header	Link-state PDU (LSP) packet header: <ul style="list-style-type: none"> • LSP ID—LSP identifier in the header. • Length—Header length, in bytes. • Allocated length—Length available for the header, in bytes. • Remaining lifetime—Remaining lifetime of the link-state PDU, in seconds. • Interface—The interface from which the LSP is received. • Estimated free bytes—Estimated number of available bytes in the LSP. • Actual free bytes—Actual number of available bytes in the LSP. • Aging timer expires in—Remaining lifetime of the LSP, in seconds. 	extensive
Packet	Link-state PDU (LSP) packet: <ul style="list-style-type: none"> • LSP ID—Identifier for the link-state packet. • Length—Packet length, in bytes. • Lifetime—Remaining lifetime, in seconds. • Checksum—Checksum of the link-state PDU. • Sequence—Sequence number of the link-state PDU. This number increments whenever the link-state PDU is updated. • Fixed length—Set length for the packet, in bytes. • Version—Protocol version. • Sysid length—Length of the system ID, in bytes. The value 0 represents 6 bytes. • Packet type—Protocol data unit (PDU) type of the LSP. • SW version—Junos OS Release number. 	extensive

Table 19: show virtual-chassis protocol database Output Fields (*continued*)

Field Name	Field Description	Level of Output
TLVs	<p>Link-state PDU (LSP) type, length, and value (TLV):</p> <ul style="list-style-type: none"> • Member ID—Identifier configured for the Virtual Chassis member router. • VC ID—Identifier assigned to the Virtual Chassis member router. • Flags—Internal flags that keep track of the member state for the purpose of mastership election in the Virtual Chassis. • Priority—Priority value associated with the role assigned to a member router in the preprovisioned Virtual Chassis configuration. For example, the priority value for the routing-engine role is 129. The priority value is used for mastership election in the Virtual Chassis. • System ID—System ID of the device associated with the Virtual Chassis port interface. The System ID is derived from the device's media access control (MAC) address. • Device ID—Identifier for the device; usually the same as the Member ID. • Neighbor Info—System ID, Virtual Chassis port interface, and metric value for VCCP neighbor. • Topology Info—System ID of the VCCP neighbor. • IRI Addr Info—Internal routing interface (IRI) IP address, which is reserved for internal communication. • Master Info—System ID of the master router in the Virtual Chassis. • Backup Info—System ID of the backup router in the Virtual Chassis. • Stable State Info—Internal state information used for mastership election in the Virtual Chassis. • Member Info—System ID, Member ID, and role of each member router in the Virtual Chassis. • Provision Info—Member ID and chassis serial number specified for each member router in the preprovisioned configuration for an MX Series Virtual Chassis. • Unknown TLV—Type and length of TLVs with unsupported content received on this device. 	extensive
number queued	Number of link-state PDUs queued on the specified Virtual Chassis port interface.	extensive

Sample Output

show virtual-chassis
protocol database
all-members brief

{master:member1-re0}

user@host> show virtual-chassis protocol database all-members brief
member0:

```
-----
LSP ID                Sequence Checksum Lifetime
001d.b510.0800.00-00   0x9eb  0xb8f1    115
b0c6.9abf.6800.00-00   0x9ee  0x8f35    116
  2 LSPs
```

member1:

```
-----
LSP ID                Sequence Checksum Lifetime
001d.b510.0800.00-00   0x9eb  0xb8f1    117
b0c6.9abf.6800.00-00   0x9ee  0x8f35    114
  2 LSPs
```

show virtual-chassis
protocol database
member 0 detail

{master:member1-re0}

user@host> show virtual-chassis protocol database member 0 detail
member0:

```
-----
001d.b510.0800.00-00 Sequence: 0x9f5, Checksum: 0x5b2b, Lifetime: 116 secs
Neighbor: b0c6.9abf.6800.00 Interface: vcp-1/3/0.32768 Metric: 15

b0c6.9abf.6800.00-00 Sequence: 0x9f8, Checksum: 0x326e, Lifetime: 117 secs
Neighbor: 001d.b510.0800.00 Interface: vcp-5/0/0.32768 Metric: 15
```

show virtual-chassis
protocol database

{master:member1-re0}

user@host> show virtual-chassis protocol database member 0 b0c6.9abf.6800 detail
member0:

member 0

b0c6.9abf.6800 detail

b0c6.9abf.6800.00-00 Sequence: 0xa06, Checksum: 0x925b, Lifetime: 117 secs
Neighbor: 001d.b510.0800.00 Interface: vcp-5/0/0.32768 Metric: 15

show virtual-chassis
protocol database
member 0 extensive

{master:member1-re0}

user@host> show virtual-chassis protocol database member 0 extensive
member0:

001d.b510.0800.00-00 Sequence: 0xa09, Checksum: 0xa696, Lifetime: 116 secs
Neighbor: b0c6.9abf.6800.00 Interface: vcp-1/3/0.32768 Metric: 15

Header: LSP ID: 001d.b510.0800.00-00, Length: 804 bytes
Allocated length: 804 bytes,
Remaining lifetime: 116 secs, Interface: 64
Estimated free bytes: 0, Actual free bytes: 0
Aging timer expires in: 116 secs

Packet: LSP ID: 001d.b510.0800.00-00, Length: 804 bytes, Lifetime : 118 secs
Checksum: 0xa696, Sequence: 0xa09,
Fixed length: 27 bytes, Version: 1, Sysid length: 0 bytes
Packet type: 18, SW version: 11.1

TLVs:

Node Info: Member ID: 1, VC ID: 5a6a.e747.8511, Flags: 3, Priority: 129
System ID: 001d.b510.0800, Device ID: 1
Unknown TLV, Type: 0, Length: 0
...
Unknown TLV, Type: 0, Length: 0
Unknown TLV, Type: 1, Length: 1
Neighbor Info: b0c6.9abf.6800.00, Interface: vcp-1/3/0.32768, Metric: 15
Topology Info: System ID: 001d.b510.0800,
Topology Info: System ID: b0c6.9abf.6800,
IRI Addr Info: IP Address: 128.0.0.1,
IRI Addr Info: IP Address: 128.0.0.4,
IRI Addr Info: IP Address: 128.0.0.5,
IRI Addr Info: IP Address: 128.0.0.6,
IRI Addr Info: IP Address: 128.0.0.17,
Master Info: System ID: 001d.b510.0800
Backup Info: System ID: b0c6.9abf.6800
Stable State Info: Master ID: 001d.b510.0800, Backup ID: b0c6.9abf.6800
Member Info: System ID: b0c6.9abf.6800, Member ID: 0 Member role: Backup
System ID: b0c6.9abf.6800, Device ID: 0
Member Info: System ID: 001d.b510.0800, Member ID: 1 Member role: Master
System ID: 001d.b510.0800, Device ID: 1
Provision Info: Member ID: 1 Serial Number: JN10C78D1AFC,
Provision Info: Member ID: 0 Serial Number: JN115FDADAFB,
Unknown TLV, Type: 24, Length: 1
Unknown TLV, Type: 28, Length: 56

1 queued :

Send PSN on vcp-5/0/0.32768 for 00:00:01

b0c6.9abf.6800.00-00 Sequence: 0xa0d, Checksum: 0x82d2, Lifetime: 118 secs
Neighbor: 001d.b510.0800.00 Interface: vcp-5/0/0.32768 Metric: 15

Header: LSP ID: b0c6.9abf.6800.00-00, Length: 808 bytes
Allocated length: 1400 bytes,
Remaining lifetime: 118 secs, Interface: 0
Estimated free bytes: 546, Actual free bytes: 592

Aging timer expires in: 118 secs

Packet: LSP ID: b0c6.9abf.6800.00-00, Length: 808 bytes, Lifetime : 118 secs
 Checksum: 0x82d2, Sequence: 0xa0d,
 Fixed length: 27 bytes, Version: 1, Sysid length: 0 bytes
 Packet type: 18, SW version: 11.1

TLVs:

Node Info: Member ID: 0, VC ID: 5a6a.e747.8511, Flags: 5, Priority: 129
 System ID: b0c6.9abf.6800, Device ID: 0

Unknown TLV, Type: 0, Length: 0

...

Unknown TLV, Type: 0, Length: 0

Unknown TLV, Type: 1, Length: 1

Neighbor Info: 001d.b510.0800.00, Interface: vcp-5/0/0.32768, Metric: 15

Topology Info: System ID: 001d.b510.0800,

Topology Info: System ID: b0c6.9abf.6800,

IRI Addr Info: IP Address: 128.0.0.1,

IRI Addr Info: IP Address: 128.0.0.4,

IRI Addr Info: IP Address: 128.0.0.5,

IRI Addr Info: IP Address: 128.0.0.6,

IRI Addr Info: IP Address: 128.0.0.17,

IRI Addr Info: IP Address: 128.0.0.21,

Master Info: System ID: 001d.b510.0800

Backup Info: System ID: b0c6.9abf.6800

Stable State Info: Master ID: 001d.b510.0800, Backup ID: b0c6.9abf.6800

Member Info: System ID: b0c6.9abf.6800, Member ID: 0 Member role: Backup

System ID: b0c6.9abf.6800, Device ID: 0

Member Info: System ID: 001d.b510.0800, Member ID: 1 Member role: Master

System ID: 001d.b510.0800, Device ID: 1

Provision Info: Member ID: 1 Serial Number: JN10C78D1AFC,

Provision Info: Member ID: 0 Serial Number: JN115FDADAFB,

Unknown TLV, Type: 24, Length: 1

Unknown TLV, Type: 28, Length: 56

1 queued :

Retransmit on vcp-5/0/0.32768 for 00:00:01

show virtual-chassis protocol interface (MX Series Virtual Chassis)

Syntax	show virtual-chassis protocol interface <(brief detail)> <interface-name> <(all-members local member <i>member-id</i>)>
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display Virtual Chassis Control Protocol (VCCP) information about Virtual Chassis port interfaces in an MX Series Virtual Chassis. You can issue the show virtual-chassis protocol interface command from the console of either member router in the Virtual Chassis.
Options	<p>brief detail—(Optional) Display the specified level of output. Using the brief option is equivalent to issuing the command with no options (the default). The detail option provides more output than the brief option.</p> <p>all-members—(Optional) Display VCCP information about Virtual Chassis port interfaces for both member routers in a Virtual Chassis. This is the default behavior if no options are specified.</p> <p>interface-name—(Optional) Display VCCP information about Virtual Chassis port interfaces for the specified Virtual Chassis port, in the format vcp-slot/pic/port.logical-unit-number.</p> <p>local—(Optional) Display VCCP information about Virtual Chassis port interfaces for the member router on which you are issuing the command.</p> <p>member member-id—(Optional) Display VCCP information about Virtual Chassis port interfaces for the specified member router. Replace member-id with the value 0 or 1.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database on page 132
List of Sample Output	show virtual-chassis protocol interface brief all-members on page 162 show virtual-chassis protocol interface detail all-members on page 162 show virtual-chassis protocol interface detail local on page 162
Output Fields	Table 20 on page 160 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 20: show virtual-chassis protocol interface Output Fields

Field Name	Field Description	Level of Output
membern	Member ID of the Virtual Chassis member router for which output is displayed.	All levels

Table 20: show virtual-chassis protocol interface Output Fields (*continued*)

Field Name	Field Description	Level of Output
Interface	Name of the Virtual Chassis port interface, in the format <i>vcp-slot/pic/port.logical-unit-number</i> .	brief
State	State of the Virtual Chassis port interface: <ul style="list-style-type: none"> • Up—The interface is up. • Down—The interface is down. 	brief
Metric	Metric value for this Virtual Chassis port interface.	All levels
<i>vcp-slot/ pic/port. logical-unit-number</i>	Name of the Virtual Chassis port interface.	detail
Index	Interface index number assigned by the Junos OS software.	detail
State	Internal implementation information.	detail
LSP interval	Interval, in milliseconds, between link-state protocol data units (PDUs) sent from the interface.	detail
type Hello padding	Type of hello padding: <ul style="list-style-type: none"> • Adaptive—On point-to-point connections, the hello packets are padded from the initial detection of a new neighbor until the neighbor verifies the adjacency as Up in the adjacency state type, length, and value (TLV). If the neighbor does not support the adjacency state TLV, then padding continues. On LAN connections, padding starts from the initial detection of a new neighbor until there is at least one active adjacency on the interface. • Loose—(Default) The hello packet is padded from the initial detection of a new neighbor until the adjacency transitions to the Up state. • Strict—Padding is performed on all interface types and for all adjacency states, and is continuous. 	detail
Adjacencies	Number of adjacencies established on this Virtual Chassis port interface.	detail
Hello(s)	Hello interval for the Virtual Chassis port interface.	detail
Hold(s)	Hold time for the Virtual Chassis port interface.	detail

Sample Output

**show virtual-chassis
protocol interface brief
all-members**

```
{master:member1-re0}  
user@host> show virtual-chassis protocol interface brief all-members  
member0:
```

```
-----  
IS-IS interface database:  
Interface          State      Metric  
vcp-5/0/0.32768    Up         15
```

member1:

```
-----  
IS-IS interface database:  
Interface          State      Metric  
vcp-1/3/0.32768    Up         15
```

**show virtual-chassis
protocol interface
detail all-members**

```
{master:member1-re0}  
user@host> show virtual-chassis protocol interface detail all-members  
member0:
```

```
-----  
IS-IS interface database:  
vcp-5/0/0.32768  
  Index: 64, State: 0x46  
  LSP interval: 100 ms, Loose Hello padding  
  Adjacencies Metric Hello (s) Hold (s)n      1    15      3    60
```

member1:

```
-----  
IS-IS interface database:  
vcp-1/3/0.32768  
  Index: 64, State: 0x86  
  LSP interval: 100 ms, Loose Hello padding  
  Adjacencies Metric Hello (s) Hold (s)n      1    15      3    60
```

**show virtual-chassis
protocol interface
detail local**

```
{master:member1-re0}  
user@host> show virtual-chassis protocol interface detail local
```

```
IS-IS interface database:  
vcp-1/3/0.32768  
  Index: 64, State: 0x46  
  LSP interval: 100 ms, Loose Hello padding  
  Adjacencies Metric Hello (s) Hold (s)n      1    15      3    60
```

show virtual-chassis protocol route (MX Series Virtual Chassis)

Syntax	show virtual-chassis protocol route < <i>destination-id</i> > <(all-members local member <i>member-id</i>)>
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display the Virtual Chassis Control Protocol (VCCP) unicast and multicast routing tables for an MX Series Virtual Chassis. You can issue the show virtual-chassis protocol route command from the console of either member router in the Virtual Chassis.
Options	<p>all-members—(Optional) Display the VCCP unicast and multicast routing tables for both member routers in a Virtual Chassis configuration. This is the default behavior if no options are specified.</p> <p><i>destination-id</i>—(Optional) Display the VCCP unicast and multicast routing tables to the destination with the specified system ID.</p> <p>local—(Optional) Display the VCCP unicast and multicast routing tables for the member router on which you are issuing the command.</p> <p>member <i>member-id</i>—(Optional) Display the VCCP unicast and multicast routing tables for the specified member router. Replace <i>member-id</i> with the value 0 or 1.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> Viewing Virtual Chassis Control Protocol Routing Tables on page 132
List of Sample Output	show virtual-chassis protocol route all-members on page 165 show virtual-chassis protocol route member 0 001d.b510.0800 (For Specific Member ID and Destination ID) on page 165
Output Fields	Table 21 on page 163 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 21: show virtual-chassis protocol route Output Fields

Field Name	Field Description
membern	Member ID of the Virtual Chassis member router for which output is displayed.
Dev	System ID of the device (member router) that stores the VCCP routing tables. The System ID is derived from the router's media access control (MAC) address.
ucast routing table	VCCP unicast routing table.
mcast routing table	VCCP multicast routing table.

Table 21: show virtual-chassis protocol route Output Fields (*continued*)

Field Name	Field Description
Current version	Version of the shortest-path-first (SPF) algorithm that generated the VCCP unicast or multicast routing table.
System ID	System ID of the device, derived from the device's MAC address.
Version	Version of the SPF algorithm that generated this route in the VCCP unicast or multicast routing table.
Metric	Metric value required to reach this device.
Interface	Name of the Virtual Chassis port interface (in the format <i>vcp-slot/pic/port.logical-unit-number</i>) that interconnects the devices.
Via	MAC address of the next-hop device, if applicable.

Sample Output

show virtual-chassis
protocol route
all-members

```
{master:member1-re0}
user@host> show virtual-chassis protocol route all-members
member0:
-----
Dev b0c6.9abf.6800 ucast routing table          Current version: 17
-----
System ID      Version  Metric Interface  Via
001d.b510.0800    17      15 vcp-5/0/0.32768 001d.b510.0800
b0c6.9abf.6800    17      0
Dev b0c6.9abf.6800 mcast routing table          Current version: 17
-----
System ID      Version  Metric Interface  Via
001d.b510.0800    17
b0c6.9abf.6800    17      vcp-5/0/0.32768
member1:
-----
Dev 001d.b510.0800 ucast routing table          Current version: 17
-----
System ID      Version  Metric Interface  Via
001d.b510.0800    17      0
b0c6.9abf.6800    17      15 vcp-1/3/0.32768 b0c6.9abf.6800
Dev 001d.b510.0800 mcast routing table          Current version: 17
-----
System ID      Version  Metric Interface  Via
001d.b510.0800    17      vcp-1/3/0.32768
b0c6.9abf.6800    17
```

show virtual-chassis
protocol route member
0 001d.b510.0800
(For Specific Member
ID and Destination ID)

```
{master:member1-re0}
user@host> show virtual-chassis protocol route member 0 001d.b510.0800
member0:
-----
Dev b0c6.9abf.6800 ucast routing table          Current version: 17
-----
System ID      Version  Metric Interface  Via
001d.b510.0800    17      15 vcp-5/0/0.32768 001d.b510.0800
b0c6.9abf.6800    17      0
Dev b0c6.9abf.6800 mcast routing table          Current version: 17
-----
System ID      Version  Metric Interface  Via
001d.b510.0800    17
b0c6.9abf.6800    17      vcp-5/0/0.32768
```

show virtual-chassis protocol statistics (MX Series Virtual Chassis)

Syntax	show virtual-chassis protocol statistics < <i>interface-name</i> > <(all-members local member <i>member-id</i>)>
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display Virtual Chassis Control Protocol (VCCP) statistics for one or both member routers, or for a specified Virtual Chassis port interface, in an MX Series Virtual Chassis. You can issue the show virtual-chassis protocol statistics command from the console of either member router in the Virtual Chassis.
Options	<p>all-members—(Optional) Display VCCP statistics for both member routers in a Virtual Chassis configuration. This is the default behavior if no options are specified.</p> <p><i>interface-name</i>—(Optional) Display VCCP statistics for the specified Virtual Chassis port interface, in the format vcp-slot/pic/port.logical-unit-number.</p> <p>local—(Optional) Display VCCP statistics for the member router on which you are issuing the command.</p> <p>member <i>member-id</i>—(Optional) Display VCCP statistics for the specified member router. Replace <i>member-id</i> with the value 0 or 1.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> Viewing Virtual Chassis Control Protocol Statistics for Member Routers and Virtual Chassis Ports on page 133
List of Sample Output	show virtual-chassis protocol statistics all-members on page 168 show virtual-chassis protocol statistics vcp-1/3/0.32768 member 1 (For Specific Virtual Chassis Port Interface and Member ID) on page 168
Output Fields	Table 22 on page 166 lists the output fields for the show virtual-chassis protocol statistics command. Output fields are listed in the approximate order in which they appear.

Table 22: show virtual-chassis protocol statistics Output Fields

Field Name	Field Description
membern	Member ID of the Virtual Chassis member router for which output is displayed.
PDU type	Type of protocol data unit (PDU).
Received	Number of PDUs received since VCCP started or since the statistics were set to zero.
Processed	Number of PDUs received minus the number of PDUs dropped.
Drops	Number of PDUs dropped.

Table 22: show virtual-chassis protocol statistics Output Fields (*continued*)

Field Name	Field Description
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	Total number of PDUs received since VCCP started or since the statistics were set to zero.
Sent	Total number of PDUs transmitted since VCCP started or since the statistics were set to zero.
LSP queue length	Number of link-state PDUs waiting in the queue to be processed.
Drops	Number of link-state PDUs dropped.
SPF runs	Number of shortest-path-first (SPF) calculations performed.
Fragments rebuilt	Number of link-state PDU fragments computed by the local system.
LSP regenerations	Number of regenerated link-state PDUs. A link-state PDU is regenerated when the PDU nears the end of its lifetime and has not changed.
Purges initiated	Number of purges initiated by the software. A purge is initiated when the software determines that it must remove a link-state PDU from the network.

Sample Output

```
show virtual-chassis
protocol statistics
all-members
```

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

```
user@host> show virtual-chassis protocol statistics all-members
member0:
```

```
-----
IS-IS statistics for b0c6.9abf.6800:
```

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	2937	2937	0	2934	0
HELLO	2913	2913	0	2922	0
CSNP	1	1	0	1	0
PSNP	2916	2916	0	2925	0
Unknown	0	0	0	0	0
Totals	8767	8767	0	8782	0

```
Total packets received: 8767 Sent: 8782
```

```
LSP queue length: 0 Drops: 0
SPF runs: 17
Fragments rebuilt: 2955
LSP regenerations: 14
Purges initiated: 0
```

```
member1:
```

```
-----
IS-IS statistics for 001d.b510.0800:
```

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	2934	2934	0	2937	0
HELLO	2922	2922	0	2914	0
CSNP	1	1	0	1	0
PSNP	2925	2925	0	2916	0
Unknown	0	0	0	0	0
Totals	8782	8782	0	8768	0

```
Total packets received: 8782 Sent: 8768
```

```
LSP queue length: 0 Drops: 0
SPF runs: 17
Fragments rebuilt: 2953
LSP regenerations: 11
Purges initiated: 0
```

```
show virtual-chassis
protocol statistics
vcp-1/3/0.32768
member1 (For Specific
Virtual Chassis Port
```

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

```
user@host> show virtual-chassis protocol statistics vcp-1/3/0.32768 member 1
member1:
```

```
-----
vcp-1/3/0.32768
```


Interface and Member
ID)

IS-IS statistics for 001d.b510.0800:

PDU type	Received	Processed	Drops	Sent	Rexmit
LSP	3013	3013	0	3016	0
HELLO	3001	3001	0	2993	0
CSNP	1	1	0	1	0
PSNP	3003	3003	0	2994	0
Unknown	0	0	0	0	0
Totals	9018	9018	0	9004	0

Total packets received: 9018 Sent: 9004

show virtual-chassis status (MX Series Virtual Chassis)

Syntax	show virtual-chassis status
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display information about the status of both member routers in an MX Series Virtual Chassis configuration. You can issue the show virtual-chassis status command from the console of either member router in the Virtual Chassis.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Verifying the Status of Virtual Chassis Member Routers on page 129 • Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33 • Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59
List of Sample Output	show virtual-chassis status on page 171
Output Fields	Table 23 on page 170 lists the output fields for the show virtual-chassis status command. Output fields are listed in the approximate order in which they appear.

Table 23: show virtual-chassis status Output Fields

Field Name	Field Description
Virtual Chassis ID	Assigned ID that applies to the entire Virtual Chassis configuration.
Member ID	Member ID assigned in the preprovisioned Virtual Chassis configuration, and the Flexible PIC Concentrator (FPC) slot range, including offset, for each member router in the Virtual Chassis.
Status	State of the member router: <ul style="list-style-type: none"> • Prsnt—Router is currently connected to the Virtual Chassis. • NotPrsnt—Router is not currently connected to the Virtual Chassis.
Serial No	Serial number of the member router.
Model	Model number of the member router.
Mastership priority	Metric used by the Virtual Chassis software for the mastership election algorithm. This value is assigned by the software and is not configurable in the current release.
Role	Role of the member router in the Virtual Chassis: Master or Backup . The asterisk (*) following the Role denotes the router on which the show virtual-chassis status command was issued.

Table 23: show virtual-chassis status Output Fields (*continued*)

Field Name	Field Description
Neighbor List ID Interface	Member IDs and Virtual Chassis port interfaces (in the format vcp-slot/pic/port) to which this member router is connected.

Sample Output

```

show virtual-chassis status {master:member1-re0}
user@host> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: 5a6a.e747.8511

Member ID      Status  Serial No  Model  Mastership  Role  Neighbor List
priority      ID  Interface
0 (FPC  0- 11) Prsnt   JN115FDADAFB mx480    129  Backup    1 vcp-5/0/0
1 (FPC 12- 23) Prsnt   JN10C78D1AFC mx240    129  Master*   0 vcp-1/3/0

```

show virtual-chassis vc-port (MX Series Virtual Chassis)

Syntax	<code>show virtual-chassis vc-port</code> <code><(all-members local member <i>member-id</i>)></code>
Release Information	Command introduced in Junos OS Release 11.2.
Description	Display the operational status of the Virtual Chassis ports for both member routers or for a specified member router in an MX Series Virtual Chassis configuration. You can issue the show virtual-chassis vc-port command from the console of either member router in the Virtual Chassis.
Options	<p>all-members—(Optional) Display the operational status of the Virtual Chassis ports for both member routers in a Virtual Chassis configuration.</p> <p>local—(Optional) Display the operational status of the Virtual Chassis ports on the member router on which you are issuing the command. This is the default behavior if no options are specified.</p> <p>member <i>member-id</i>—(Optional) Display the operational status of the Virtual Chassis ports on the specified member router. Replace <i>member-id</i> with the value 0 or 1.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Verifying the Operation of Virtual Chassis Ports on page 129 • Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 33 • Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59
List of Sample Output	show virtual-chassis vc-port all-members on page 173 show virtual-chassis vc-port local on page 173 show virtual-chassis vc-port member 0 on page 173
Output Fields	Table 24 on page 172 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 24: show virtual-chassis vc-port Output Fields

Field Name	Field Description
membern	Member ID of the Virtual Chassis member router for which output is displayed.
Interface or Slot/PIC/Port	Location, in the format <i>slot/pic/port</i> , of the Virtual Chassis ports configured on the member router.
Type	Type of Virtual Chassis port. Configured indicates that the Virtual Chassis port is properly configured.

Table 24: show virtual-chassis vc-port Output Fields (*continued*)

Field Name	Field Description
Trunk ID	Trunk ID value assigned to a link aggregation group (LAG) formed by the Virtual Chassis. A positive number indicates that a trunk exists. The value -1 indicates that a trunk is not present.
Status	State of the Virtual Chassis port interface: Up , Down , or Absent .
Speed (mbps)	Speed, in megabits per second, of the Virtual Chassis port interface.
Neighbor ID Interface	Member IDs and Virtual Chassis port interfaces (in vcp-slot/pic/port format) that are connected to this member router.

Sample Output

show virtual-chassis
vc-port all-members

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

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

```
member0:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID Interface
5/0/0	Configured	-1	Up	10000	1 vcp-1/3/0

```
member1:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID Interface
1/3/0	Configured	-1	Up	10000	0 vcp-5/0/0

show virtual-chassis
vc-port local

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

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

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID Interface
1/3/0	Configured	-1	Up	10000	0 vcp-5/0/0

show virtual-chassis
vc-port member 0

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

```
user@host> show virtual-chassis vc-port member 0
```

```
member0:
```

Interface or Slot/PIC/Port	Type	Trunk ID	Status	Speed (mbps)	Neighbor ID Interface
5/0/0	Configured	-1	Up	10000	1 vcp-1/3/0

PART 4

Troubleshooting

- [Acquiring Troubleshooting Information on page 177](#)
- [Troubleshooting Configuration Statements on page 185](#)

CHAPTER 12

Acquiring Troubleshooting Information

- [Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 178](#)
- [Configuring the Name of the Virtual Chassis Trace Log File on page 179](#)
- [Configuring Characteristics of the Virtual Chassis Trace Log File on page 179](#)
- [Configuring Access to the Virtual Chassis Trace Log File on page 180](#)
- [Using Regular Expressions to Refine the Output of the Virtual Chassis Trace Log File on page 181](#)
- [Configuring the Virtual Chassis Operations to Trace on page 182](#)

Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers

The Junos OS trace feature tracks Virtual Chassis operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems.

By default, tracing is disabled. When you enable the tracing operation on the router to be configured as the master (also referred to as the *protocol master*) of an MX Series Virtual Chassis, the default tracing behavior is as follows:

1. Important events are logged in a file with the name you specify in the `/var/log` directory. You cannot change the directory (`/var/log`) in which trace files are located.
2. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file.0***, then ***trace-file.1***, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

You can optionally specify the maximum number of trace files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB). (For more information about how log files are created, see the *Junos OS System Log Messages Reference*.)

By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

To configure tracing of MX Series Virtual Chassis operations:

1. Configure a filename for the trace log.
See [“Configuring the Name of the Virtual Chassis Trace Log File” on page 179](#).
2. (Optional) Configure characteristics of the trace log file.
See [“Configuring Characteristics of the Virtual Chassis Trace Log File” on page 179](#).
3. (Optional) Configure user access to the trace log file.
See [“Configuring Access to the Virtual Chassis Trace Log File” on page 180](#).
4. (Optional) Refine the output of the trace log file.
See [“Using Regular Expressions to Refine the Output of the Virtual Chassis Trace Log File” on page 181](#).
5. Configure flags to specify the Virtual Chassis operations that you want to trace.
See [“Configuring the Virtual Chassis Operations to Trace” on page 182](#).

Related Documentation

- [Configuring Preprovisioned Member Information for a Virtual Chassis on page 44](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Configuring the Name of the Virtual Chassis Trace Log File

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

To configure the filename for tracing MX Series Virtual Chassis operations:

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

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

Related Documentation

- [Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 178](#)
- [Configuring Preprovisioned Member Information for a Virtual Chassis on page 44](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Configuring Characteristics of the Virtual Chassis Trace Log File

You can optionally configure the following characteristics of the trace log file for an MX Series Virtual Chassis:

- **Maximum number of trace files**—When a trace file named *trace-file* reaches its maximum size, it is renamed *trace-file.0*, then *trace-file.1*, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. You can optionally specify the maximum number of trace files to be from 2 through 1000. If you specify a maximum number of files with the **files** option, you must also specify a maximum file size with the **size** option.
- **Maximum trace file size**—You can configure the maximum trace file size to be from 10 KB through 1 gigabyte (GB). If you specify a maximum file size with the **size** option, you must also specify a maximum number of files with the **files** option.
- **Timestamp**—By default, timestamp information is placed at the beginning of each line of trace output. You can optionally prevent placement of a timestamp on any trace log file.
- **Appending or replacing the trace file**—By default, the router appends new information to an existing trace file. You can optionally specify that the router replace an existing trace file instead of appending information to it.

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

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

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

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

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

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

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

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

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

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

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

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

Related Documentation

- [Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 178](#)
- [Configuring Preprovisioned Member Information for a Virtual Chassis on page 44](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Configuring Access to the Virtual Chassis Trace Log File

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

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

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

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

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

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

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

```
user@host# set traceoptions file filename no-world-readable
```

- Related Documentation**
- [Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 178](#)
 - [Configuring Preprovisioned Member Information for a Virtual Chassis on page 44](#)
 - [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

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

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

To refine the output of the trace log file:

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

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

```
user@host# set traceoptions file filename match regular-expression
```

- Related Documentation**
- [Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 178](#)
 - [Configuring Preprovisioned Member Information for a Virtual Chassis on page 44](#)
 - [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

Configuring the Virtual Chassis Operations to Trace

By default, the router logs only important events. You can specify which operations to trace for an MX Series Virtual Chassis by including specific tracing flags when you configure tracing. [Table 25 on page 182](#) describes the flags that you can include.

Table 25: Tracing Flags for MX Series Virtual Chassis

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

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

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

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

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

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

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

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

**Related
Documentation**

- [Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 178](#)
- [Configuring Preprovisioned Member Information for a Virtual Chassis on page 44](#)
- [Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59](#)

CHAPTER 13

Troubleshooting Configuration Statements

traceoptions (MX Series Virtual Chassis)

Syntax	<pre>traceoptions { file <i>filename</i> <files <i>number</i>> <match <i>regular-expression</i>> <no-stamp> <replace> <size <i>maximum-file-size</i>> <world-readable no-world-readable>; flag <i>flag</i> <detail> <disable> <receive> <send>; }</pre>
Hierarchy Level	[edit virtual-chassis]
Release Information	Statement introduced in Junos OS Release 11.2.
Description	Define tracing operations for the MX Series Virtual Chassis configuration.
Default	Tracing operations are disabled.
Options	<p>detail—(Optional) Generate detailed trace information for a flag.</p> <p>disable—(Optional) Disable a flag.</p> <p>file <i>filename</i>—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 <code>/var/log</code>.</p> <p>files <i>number</i>—(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.</p> <p>Range: 2 through 1000</p> <p>Default: 3 files</p> <p>flag <i>flag</i>—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:</p> <ul style="list-style-type: none">• all—All tracing operations.• auto-configuration—Trace Virtual Chassis ports that have been automatically configured.• csn—Trace Virtual Chassis complete sequence number (CSN) packets.• error—Trace Virtual Chassis errored packets.• graceful-restart—Trace Virtual Chassis graceful restart events.• hello—Trace Virtual Chassis hello packets.• krt—Trace Virtual Chassis kernel routing table (KRT) events.• lsp—Trace Virtual Chassis link-state packets.• lsp-generation—Trace Virtual Chassis link-state packet generation.• me—Trace Virtual Chassis mastership election (ME) events.

- **normal**—Trace normal events.
- **packets**—Trace Virtual Chassis packets.
- **parse**—Trace reading of the configuration.
- **psn**—Trace partial sequence number (PSN) packets.
- **route**—Trace Virtual Chassis routing information.
- **spf**—Trace Virtual Chassis shortest-path-first (SPF) events.
- **state**—Trace Virtual Chassis state transitions.
- **task**—Trace Virtual Chassis task operations.

match *regular-expression*—(Optional) Refine the output to include lines that contain the regular expression.

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 instead of appending information to it.

send—(Optional) Trace transmitted packets.

size *maximum-file-size*—(Optional) Maximum size of each trace file. By default, the number entered is treated as bytes. Alternatively, you can include a suffix to the number to indicate kilobytes (KB), megabytes (MB), or gigabytes (GB). If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: *sizek* to specify KB, *sizem* to specify MB, or *sizeg* to specify GB

Range: 10240 through 1073741824

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

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

Related Documentation	<ul style="list-style-type: none"> • Configuring Preprovisioned Member Information for a Virtual Chassis on page 44 • Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 178 • Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 59
------------------------------	---

PART 5

Index

- [Index on page 191](#)

Index

Symbols

#, comments in configuration statements.....	xiv
(), in syntax descriptions.....	xiv
< >, in syntax descriptions.....	xiv
[], in configuration statements.....	xiv
{ }, in configuration statements.....	xiv
(pipe), in syntax descriptions.....	xiv

B

braces, in configuration statements.....	xiv
brackets	
angle, in syntax descriptions.....	xiv
square, in configuration statements.....	xiv

C

command forwarding, MX Series Virtual Chassis.....	118
comments, in configuration statements.....	xiv
configuration examples, Virtual Chassis	
configuring.....	59
CoS on Virtual Chassis ports.....	99
deleting.....	73
replacing a Routing Engine.....	90
upgrading.....	85
configuration groups for MX Series Virtual Chassis.....	42
configuration guidelines, Virtual Chassis	
CoS on Virtual Chassis ports.....	30
Virtual Chassis ports.....	10
conventions	
text and syntax.....	xiii
curly braces, in configuration statements.....	xiv
customer support.....	xv
contacting JTAC.....	xv

D

documentation	
comments on.....	xv

E

examples, configuration See configuration examples	
--	--

F

font conventions.....	xiii
-----------------------	------

G

graceful Routing Engine switchover	
MX Series Virtual Chassis.....	17, 46, 55

L

license requirements for MX Series Virtual Chassis.....	40
---	----

M

manuals	
comments on.....	xv
mastership election, Virtual Chassis.....	15
member statement.....	105
MX Series Virtual Chassis commands	
show virtual-chassis protocol adjacency.....	150

N

no-split-detection statement.....	106
nonstop active routing	
MX Series Virtual Chassis.....	46

P

parentheses, in syntax descriptions.....	xiv
preprovisioned statement.....	107

R

request virtual-chassis member-id delete	
command.....	136
request virtual-chassis member-id set	
command.....	137
request virtual-chassis routing-engine master	
switch command.....	138
request virtual-chassis vc-port delete	
command.....	140
request virtual-chassis vc-port set command.....	142
role statement.....	108
roles, Virtual Chassis.....	13, 54

S

serial-number statement.....	109
show virtual-chassis active-topology	
command.....	146
show virtual-chassis device-topology	
command.....	148
show virtual-chassis protocol adjacency	
command.....	150

show virtual-chassis protocol database	
command.....	154
show virtual-chassis protocol interface	
command.....	160
show virtual-chassis protocol route command.....	163
show virtual-chassis protocol statistics	
command.....	166
show virtual-chassis status command.....	170
show virtual-chassis vc-port command.....	172
SNMP	
using with Virtual Chassis.....	126
split detection, Virtual Chassis.....	19, 57
support, technical See technical support	
switchover behavior, Virtual Chassis.....	17
syntax conventions.....	xiii

T

technical support	
contacting JTAC.....	xv
traceoptions statement	
Virtual Chassis.....	110, 186
tracing Virtual Chassis operations.....	178

U

upgrading Junos OS software	
MX Series Virtual Chassis.....	35

V

Virtual Chassis	
backup router.....	6
benefits.....	4
command forwarding.....	118
components.....	5
configuration	
overview.....	9
configuration examples.....	59, 73, 85, 90, 99
configuring.....	33
CoS on Virtual Chassis ports.....	25, 30
creating configuration groups.....	42
deleting.....	34
deleting Virtual Chassis ports.....	51
disabling split detection.....	57
flags, trace log.....	182
graceful Routing Engine switchover.....	55
graceful Routing Engine switchover	
readiness.....	17
graceful Routing Engine switchover,	
enabling.....	46
license requirements.....	40

line-card router.....	7
management interface, accessing.....	116
managing files on member routers.....	117
master router.....	6
mastership election.....	15
member IDs, configuring.....	47
member IDs, deleting.....	52
member routers and roles.....	5, 13
nonstop active routing, enabling.....	46
overview.....	3
preparing for the configuration.....	38
preprovisioned member information.....	44
replacing a Routing Engine.....	90
roles, global and local.....	13
roles, switching master and backup.....	54
slot numbering.....	8, 126
SNMP, using with.....	126
split detection.....	19, 57
supported platforms.....	4
switchover behavior.....	17
tracing operations of.....	178
upgrading Junos OS software.....	35
verifying neighbor reachability.....	130
verifying status and operation.....	129
viewing database information.....	131, 132
viewing routing tables.....	132
viewing statistics information.....	133
Virtual Chassis Control Protocol.....	9
Virtual Chassis port trunks.....	8
Virtual Chassis ports.....	7, 10, 49, 51
Virtual Chassis commands	
request virtual-chassis member-id delete.....	136
request virtual-chassis member-id set.....	137
request virtual-chassis routing-engine master	
switch.....	138
request virtual-chassis vc-port delete.....	140
request virtual-chassis vc-port set.....	142
show virtual-chassis active-topology.....	146
show virtual-chassis device-topology.....	148
show virtual-chassis protocol database.....	154
show virtual-chassis protocol interface.....	160
show virtual-chassis protocol route.....	163
show virtual-chassis protocol statistics.....	166
show virtual-chassis status.....	170
show virtual-chassis vc-port.....	172
Virtual Chassis ports	
configuration guidelines.....	10
configuring.....	49
CoS configuration example.....	99

CoS configuration guidelines.....	30
CoS overview.....	25
deleting.....	51
forming trunks.....	8
overview.....	7
slot numbering.....	8
Virtual Chassis statements	
member.....	105
no-split-detection.....	106
preprovisioned.....	107
role.....	108
serial-number.....	109
traceoptions.....	110, 186
virtual-chassis.....	112
virtual-chassis statement.....	112

