Juniper Networks | Engineering Simplicity

Junos® OS

Interchassis Redundancy Using Virtual Chassis User Guide for MX Series Routers

Published
2025-09-23

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

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

Table of Contents

About This Guide

Use this guide to configure a virtual chassis using MX Series routers.

1

CHAPTER

Understanding How Virtual Chassis Provides Interchassis Redundancy

IN THIS CHAPTER

Interchassis Redundancy and Virtual Chassis Overview

IN THIS SECTION

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 5G Universal Routing Platforms, 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

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 5G Universal Routing Platforms, 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 primary router* (also known as the *protocol primary*) 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 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 primary router, and establishing the interchassis routing table to route traffic within the Virtual Chassis.

NOTE: MX Series Virtual Chassis does not support Ethernet OAM, distributed inline connectivity fault management, Ethernet frame delay measurement, loss measurement, synthetic loss measurement, and Ethernet alarm indication signal (ETH-AIS).

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.

Supported Routing Platforms for MX Series Virtual Chassis

You can configure a Virtual Chassis on the following MX Series 5G Universal Routing Platforms with MPC/MIC interfaces:

- MX240 Universal Routing Platform
- MX480 Universal Routing Platform
- MX960 Universal Routing Platform
- MX2010 Universal Routing Platform
- MX2020 Universal Routing Platform
- MX10003 Universal Routing Platform

NOTE: Platform support depends on the Junos OS release in your installation.

Graceful Routing Engine switchover (GRES) and *nonstop active routing* (NSR) must be enabled on both member routers in the Virtual Chassis.

Supported Member Router Combinations

A two-member MX Series Virtual Chassis supports the member router combinations marked as Yes in Table 1 on page 4.

Table 1: MX Series Virtual Chassis Supported Member Router Combinations

Member Router Type	MX240	MX480	MX960	MX2010	MX2020	MX10003
MX240	Yes	Yes	Yes	No	No	No

Table 1: MX Series Virtual Chassis Supported Member Router Combinations *(Continued)*

Member Router Type	MX240	MX480	MX960	MX2010	MX2020	MX10003
MX480	Yes	Yes	Yes	No	No	No
MX960	Yes	Yes	Yes	Yes	Yes	No
MX2010	No	No	Yes	Yes	Yes	No
MX2020	No	No	Yes	Yes	Yes	No
MX10003	No	No	No	No	No	Yes

Routing Engine Requirements

Each member router in the Virtual Chassis must have dual Routing Engines installed, and all four Routing Engines in the Virtual Chassis must be the same model. For example, you cannot configure a Virtual Chassis if one member router has two RE-S-2000 Routing Engines installed and the other member router has two RE-S-1800 Routing Engines installed.

NOTE: For an MX Series Virtual Chassis configuration that includes an MX2020 router, all four Routing Engines in the Virtual Chassis must have at least 16 gigabytes of memory.

RELATED DOCUMENTATION

Virtual Chassis Components Overview | 7

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

2

CHAPTER

Understanding How a Virtual Chassis Works

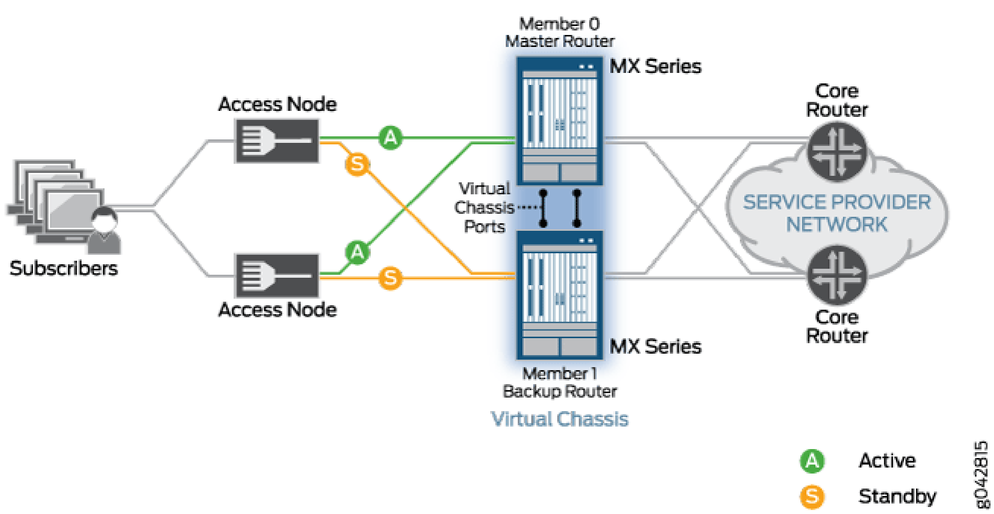
IN THIS CHAPTER

Virtual Chassis Components Overview

IN THIS SECTION

A *Virtual Chassis* configuration for MX Series 5G Universal Routing Platforms interconnects two MX Series routers into a logical system that you can manage as a single network element.Figure 1 on page 7 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 7, and covers the following topics:

Virtual Chassis Primary Router

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

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

- Manages both the primary 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 primary router by default. After the Virtual Chassis is formed with both member routers, the Virtual Chassis Control Protocol (VCCP) software runs a primary-role election algorithm to elect the primary router for the Virtual Chassis configuration.

NOTE: You cannot configure primary-role 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 primary router becomes the *backup router*, also known as the *protocol backup*. The Virtual Chassis backup router takes over the primary role of the Virtual Chassis if the primary router is unavailable, and synchronizes routing and state information with the primary router. The primary Routing Engine that resides in the Virtual Chassis backup router becomes the global backup for the Virtual Chassis.

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

- If the primary router fails or is unavailable, takes over the primary role 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 primary Routing Engine that resides in the Virtual Chassis primary router
- Relays chassis control information, such as line card presence and alarms, to the primary 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 primary 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 primary router takes a `line-card` role, and line cards (FPCs) that do not host Virtual Chassis ports go offline. This state effectively isolates the primary 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 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 7 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.

NOTE: The Junos OS software enables you to preconfigure ports that are currently unavailable for use. Although a Virtual Chassis port is unavailable for use as a standard network port, you can configure this port as a standard network port even after you configure it as a Virtual Chassis port. However, the router does not apply the configuration until you 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, a 10-Gigabit Ethernet (xe) interface, a 40-Gigabit Ethernet (et) interface, or a 100-Gigabit Ethernet (et) interface. 40-Gigabit and 100-Gigabit Virtual Chassis ports can only be configured on MPC3, MPC4, or later line cards. (Interface support depends on the Junos OS release in your installation.) You cannot configure a combination of 1-Gigabit Ethernet Virtual Chassis ports and 10-Gigabit Ethernet Virtual Chassis ports in 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. 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:

Cost = (300 * 1,000,000,000) / *port-speed*

where *port-speed* is the aggegate 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

After you configure the member ID and, optionally, slot count for each router that you want to add to an MX Series Virtual Chassis, the Routing Engines in that chassis reboot and the slots for line cards (FPCs) are renumbered. The FPC slot numbering used for each member router is based on the slot count and offsets used in the Virtual Chassis instead of the physical slot numbers where the line card is actually installed.

Table 2 on page 11 shows the valid slot count values for each supported member router type, and the slot numbering used for member 0 and member 1 when the specified slot count value is configured, either explicitly or by default.

Table 2: Slot Count and Slot Numbering for MX Series Virtual Chassis Supported Member Routers

Member Router Type	Slot Count	FPC Slot Numbers on member 0	FPC Slot Numbers on member 1
MX240	N/A	0 through 11 (no offset)	12 through 23 (offset=12)
MX480	N/A	0 through 11 (no offset)	12 through 23 (offset=12)
MX960	12 (default)	0 through 11 (no offset)	12 through 23 (offset=12)
MX960	20	0 through 19 (no offset)	20 through 39 (offset=20)

Table 2: Slot Count and Slot Numbering for MX Series Virtual Chassis Supported Member Routers *(Continued)*

Member Router Type	Slot Count	FPC Slot Numbers on member 0	FPC Slot Numbers on member 1
MX2010	12 (default)	0 through 11 (no offset)	12 through 23 (offset=12)
MX2010	20	0 through 19 (no offset)	20 through 39 (offset=20)
MX2020	20 (default)	0 through 19 (no offset)	20 through 39 (offset=20)

For example, assume that in your Virtual Chassis configuration, member 0 is an MX960 router and member 1 is an MX2010 router, with the default slot count (12) in effect on both routers. In this topology, 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 offset of 12 for member 1.

Building on this example, assume that you replace member 1 with an MX2020 member router, resulting in a Virtual Chassis with an MX960 router configured as member 0 and an MX2020 router configured as member 1. To ensure that a Virtual Chassis consisting of an MX2020 router and either an MX960 router or MX2010 router forms properly, you must explicitly set the slot count for the MX960 router or MX2010 router to 20 to match the slot count of the MX2020 router. When the FPC slots are renumbered in this topology, physical interface xe-2/2/2 on member 1 becomes xe-22/2/2 on member 1 after adding the offset of 20 for member 1. Similarly, the `show interfaces` command displays xe-22/2/2 as the interface name.

NOTE: Slot renumbering does not affect the names of Virtual Chassis ports. The Virtual Chassis port name, in the format `vcp-`*`slot/pic/port`* , is derived from the physical slot number where the port is configured. For example, vcp-3/2/0 is configured on FPC physical slot 3, PIC slot 2, port 0.

Configuration of Chassis Properties for MPCs in the Virtual Chassis

When you configure chassis properties for MPCs installed in a member router in an MX Series Virtual Chassis, keep the following points in mind:

- Statements included at the `[edit chassis member` *`member-id`* `fpc slot` *`slot-number`*`]` hierarchy level apply to the MPC (FPC) in the specified slot number only on the specified member router in the Virtual Chassis.

For example, if you issue the `set chassis member 0 fpc slot 1 power off` statement, only the MPC installed in slot 1 of member ID 0 in the Virtual Chassis is powered off.

- Statements included at the `[edit chassis fpc slot` *`slot-number`*`]` hierarchy level should be relocated to the `[edit chassis member` *`member-id`* `fpc slot` *`slot-number`*`]` hierarchy level to avoid errors.

BEST PRACTICE: To ensure that the statement you use to configure MPC chassis properties in a Virtual Chassis applies to the intended member router and MPC, always include the `member` *`member-ID`* option before the `fpc` keyword, where *`member-id`* is 0 or 1 for a two-member MX Series Virtual Chassis.

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 primary-role election algorithm to determine the Virtual Chassis primary 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 (primary 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 primary 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

Global Roles and Local Roles in a Virtual Chassis

IN THIS SECTION

In a *Virtual Chassis* configuration for MX Series 5G Universal Routing Platforms or EX9200 switches, each of the two member devices and each of the two Routing Engines in each member device has a distinct role. A *global role* defines the function of each member device 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 device, and applies locally only to that member device.

Global roles change when you switch the Virtual Chassis primary role, and both global roles and local roles change when you switch the Routing Engine primary role in one of the member devices. In addition, the **line-card** global role, though not supported in a preprovisioned configuration for a two-member MX Series or EX9200 Virtual Chassis, applies in the context of split detection behavior.

This topic describes the global roles and local roles in a MX Series or EX9200 Virtual Chassis so you can better understand how the Virtual Chassis behaves during a global primary role switch, a local Routing Engine switchover, or when split detection is enabled.

Role Name Format

The global and local role names in an MX Series or EX9200 Virtual Chassis use the following format:

VC-*GlobalRole*< *LocalRole*>

where:

- *GlobalRole* applies to the global function of the member device for the entire Virtual Chassis, and can be one of the following:
  - **M**—Virtual Chassis primary device, also referred to as the protocol primary.
  - **B**—Virtual Chassis backup device, also referred to as the protocol backup.
  - **L**—Virtual Chassis line-card device. The **line-card** role is not supported in the preprovisioned configuration for a two-member 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 device, and can be one of the following:
  - **m**—Primary Routing Engine
  - **s**—Standby Routing Engine

Global Role and Local Role Descriptions

Table 3 on page 16 describes the global roles and local roles in an MX Series or EX9200 Virtual Chassis.

Table 3: Global Roles and Local Roles

Virtual Chassis Role	Type of Role	Description
VC-P	Global	Primary device for the Virtual Chassis
VC-B	Global	Backup device for the Virtual Chassis
VC-L	Global	Line-card device for the Virtual Chassis **NOTE**: The **line-card** role is not supported in the preprovisioned configuration for a two-member MX Series or EX9200 Virtual Chassis. The **line-card** role applies only in the context of split detection behavior.
VC-Pp	Local	Primary Routing Engine in the Virtual Chassis primary device
VC-Ps	Local	Standby Routing Engine in the Virtual Chassis primary device
VC-Bp	Local	Primary Routing Engine in the Virtual Chassis backup device
VC-Bs	Local	Standby Routing Engine in the Virtual Chassis backup device
VC-Lm	Local	Primary Routing Engine in the Virtual Chassis line-card device **NOTE**: The **line-card** role is not supported in the preprovisioned configuration for a two-member MX Series or EX9200 Virtual Chassis. The **line-card** role applies only in the context of split detection behavior.

Table 3: Global Roles and Local Roles ***(Continued)***

Virtual Chassis Role	Type of Role	Description
VC-Ls	Local	Standby Routing Engine in the Virtual Chassis line-card device **NOTE**: The **line-card** role is not supported in the preprovisioned configuration for a two-member MX Series or EX9200 Virtual Chassis. The **line-card** role applies only in the context of split detection behavior.

RELATED DOCUMENTATION

Virtual Chassis Components Overview | 7
Primary-role Election in a Virtual Chassis | 25
Switching the Global Primary and Backup Roles in a Virtual Chassis Configuration | 119
Disabling Split Detection in a Virtual Chassis Configuration

Configuring a Virtual Chassis Heartbeat Connection

IN THIS SECTION

Starting in Junos OS Release 14.1, you must configure an IP-based, bidirectional “heartbeat” packet connection between the primary router and backup router in an MX Series Virtual Chassis. The heartbeat connection determines the health and availability of member routers in the Virtual Chassis.

The member routers forming this *heartbeat connection* exchange *heartbeat packets* that provide critical information about the availability and health of each member router. During a disruption or split in the Virtual Chassis configuration, the heartbeat connection prevents the member routers from changing primary role roles unnecessarily. Without the heartbeat connection, a change in primary role roles in such a situation can produce undesirable results, such as having two Virtual Chassis primary routers or no Virtual Chassis primary router.

Benefits of Configuring a Virtual Chassis Heartbeat Connection

Configuring a Virtual Chassis heartbeat connection provides the following benefits for an MX Series Virtual Chassis:

- Improved resiliency during failure scenarios

  Configuring the heartbeat connection improves resiliency of the Virtual Chassis in the event of an adjacency disruption or split caused by a failure of the Virtual Chassis port interfaces, or when one of the member routers goes out of service. If the heartbeat connection detects that the Virtual Chassis primary router (VC-P) is still operating and able to respond during a split, the software maintains primary role on the existing VC-P, isolates the Virtual Chassis backup router (VC-B) until the Virtual Chassis recovers, and resumes the backup role on the VC-B when the Virtual Chassis forms again. As a result, the heartbeat connection prevents the member routers from unnecessarily changing primary role roles, which consumes system resources and causes unexpected and undesirable results.

  When the VC-B is isolated during a disruption, the software immediately restarts all line cards and powers off all network ports until the disruption is resolved and the Virtual Chassis forms again. This behavior supports network applications with external equipment that requires a physical link-down condition to switch the traffic paths to other connections.

- Enhanced primary-role election process

  The Virtual Chassis Control Protocol (VCCP) controls primary-role election in a Virtual Chassis. When you configure the heartbeat connection in an MX Series Virtual Chassis, the VCCP software assesses the health information collected from the heartbeat connection to help determine which member router should become the global primary (VC-P) in the event of an adjacency disruption or split. When the heartbeat connection detects that the peer member router is responsive, the VCCP software suppresses unnecessary changes in primary role roles.

  By contrast, when the heartbeat connection is *not* configured, the VCCP software does not have this additional health information when determining the appropriate primary role roles after a disruption or split.

- Ability to easily view and clear statistics related to the heartbeat connection

Operational commands for the Virtual Chassis enable you to display the status of the heartbeat connection, review detailed statistics and latency measurements related to the heartbeat connection, and clear heartbeat-related statistics counters and timestamp fields for one or both member routers.

Configuration Requirements for the Heartbeat Connection

To establish a heartbeat connection for an MX Series Virtual Chassis, you must configure a secure and reliable route between the primary router and backup router for the exchange of TCP/IP heartbeat packets. Specifically, you must ensure that the primary Routing Engine in the Virtual Chassis backup router (VC-Bp) can make a TCP/IP connection to the `master-only` IP address of the primary Routing Engine in the Virtual Chassis primary router (VC-Pp).

The following additional requirements apply when you configure the heartbeat connection:

- Configure the heartbeat connection only between Virtual Chassis member routers eligible to become the Virtual Chassis primary router, also known as the *protocol primary* or *global primary*.

  In a two-member MX Series Virtual Chassis configuration, you assign the `routing-engine` role to each router as part of the preprovisioned configuration. The `routing-engine` role enables the router to function either as the primary router or backup router of the Virtual Chassis as needed. As a result, you can configure the heartbeat connection between both member routers in a two-member MX Series Virtual Chassis configuration.

- Use the router's Ethernet management interface (`fxp0`) as the heartbeat path.

  The management interface is generally available earlier than the line card interfaces, and is typically connected to a more secure network than the other interfaces.

- Configure a `master-only` IP address for the `fxp0` management interface to ensure consistent access to the VC-Pp, regardless of which Routing Engine is currently active.

  The `master-only` address is active only on the management interface for the VC-Pp. During a switchover, the `master-only` address moves to the new Routing Engine currently functioning as the VC-Pp.

- Ensure TCP connectivity between the VC-Pp and VC-Bp member routers

  The Virtual Chassis heartbeat connection opens a proprietary TCP port numbered 33087 on the VC-Pp to listen for heartbeat messages. If your network design includes firewalls or filters, make sure the network allows traffic between TCP port 33087 on the VC-Pp and the dynamically allocated TCP port on the VC-Bp.

- When using a heartbeat connection, do not configure the `no-split-detection` statement as part of the preprovisioned Virtual Chassis configuration.

The no-split-detection statement suppresses any action when a split is detected in the Virtual Chassis. Using the no-split-detection statement is prohibited when you configure a heartbeat connection, and the software prevents you from configuring both the no-split-detection and heartbeat-address statements at the same time. If you attempt to do so, the software displays an error message and causes the commit operation to fail.

In a two-member MX Series Virtual Chassis, you can configure a heartbeat connection with both member routers in the same subnet, or with each member router in a different subnet. Table 4 on page 20 summarizes the important differences between the configuration procedures for member routers in the same subnet and member routers in different subnets.

Table 4: Comparison of Heartbeat Connection Configuration Tasks for Member Routers in Same Subnet and Member Routers in Different Subnets

Task	Heartbeat Connection for Member Routers in *Same Subnet*	Heartbeat Connection for Member Routers in *Different Subnets*
Configure the master-only IP address for fxp0 management interface.	Configure the same fxp0 master-only IP address for all four member Routing Engines.	Configure two different master-only IP addresses for the fxp0 management interface: one address for the subnet in which the Virtual Chassis primary router resides, and one for the subnet in which the backup router resides.
Configure a network path for the heartbeat connection.	Provide a path for the member routers to reach each other by means of a TCP/IP connection. For example, in a Virtual Chassis with member routers in the same subnet, you can use the router's default gateway. Alternatively, you can create a global static route as described in "Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in the Same Subnet" on page 259.	Provide a path for the member routers to reach each other by means of a TCP/IP connection. In a Virtual Chassis with member routers in different subnets, you must ensure that both member routers can reach each other's network. For example, you can create static routes to both subnets on each member Routing Engine, as described in "Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in Different Subnets" on page 274.

Table 4: Comparison of Heartbeat Connection Configuration Tasks for Member Routers in Same Subnet and Member Routers in Different Subnets *(Continued)*

Task	Heartbeat Connection for Member Routers in *Same Subnet*	Heartbeat Connection for Member Routers in *Different Subnets*
Configure the heartbeat address to establish the heartbeat connection.	Configure a single (global) `master-only` IP address for the `fxp0` management interface as the heartbeat address to establish the connection.	Configure a heartbeat address for each member Routing Engine to cross-connect to the `master-only` IP address for the corresponding Routing Engine in the other subnet. For example, assume that `member0-re0` and `member0-re1` reside in subnet 10.4.0.0, and `member1-re0` and `member1-re1` reside in subnet 10.5.0.0. In this configuration, you would set the heartbeat address for `member0-re0` to the `master-only` IP address for `member1-re0` to cross-connect `member0-re0` and `member1-re0`. You would cross-connect `member0-re1` and `member1-re1` in a similar manner.

How the Heartbeat Connection Works

When the Virtual Chassis is operating properly, the heartbeat connection periodically sends heartbeat packets over the TCP/IP path between the primary Routing Engine in the Virtual Chassis primary router and the primary Routing Engine in the Virtual Chassis backup router.

When an adjacency disruption or split is detected in the Virtual Chassis, each member router sends a final heartbeat message to determine whether the other member is able to respond, and stops sending additional periodic messages until the Virtual Chassis forms again. The other member must respond to the heartbeat message within the default heartbeat timeout period (2 seconds), or within a configured heartbeat timeout period in the range 1 through 60 seconds. To determine the time period that elapses in your network between transmission of a heartbeat request message and receipt of a heartbeat response message, you can issue the `show virtual-chassis heartbeat detail` command to view the number of seconds reported in the `Maximum latency` and `Minimum latency` fields.

BEST PRACTICE: If your network is congested or has a round-trip latency that exceeds 2 seconds, we recommend that you increase the value of the heartbeat timeout period to account for this delay during a Virtual Chassis adjacency disruption or split.

Heartbeat Connection and Virtual Chassis Failure Conditions

Configuring the heartbeat connection prevents unnecessary primary role changes between the Virtual Chassis member routers when an adjacency disruption or split occurs. Table 5 on page 22 describes the effects on primary role for common failure conditions when you enable the heartbeat connection in a two-member MX Series Virtual Chassis.

Table 5: Effect of Heartbeat Connection on Common Virtual Chassis Failure Conditions

Failure Condition	Result on Virtual Chassis Primary Router (VC-P)	Result on Virtual Chassis Backup Router (VC-B)
Virtual Chassis port interfaces go down.	Retains VC-P role.	If the VC-P is in service but the Virtual Chassis port interfaces are down, the VC-B goes offline after the heartbeat timeout period expires because the Routing Engine state is invalid.
VC-P chassis fails.	Goes out of service.	Becomes VC-P.
VC-B chassis fails.	Retains VC-P role.	Goes out of service.
Heartbeat connection fails.	Retains VC-P role.	Retains VC-B role.

In all cases except when the VC-P chassis fails, primary role of the Virtual Chassis is maintained on the existing VC-Pp if the heartbeat connection detects that the VC-P is still operating and able to respond during a split. Preventing an unnecessary role change minimizes the system load caused by a protocol primary role switch, and reduces the likelihood of unpredictable results.

Lack of Virtual Chassis Heartbeat connection and VCP adjacency loss is a double-fault condition that effectively returns to “no-split-detection” behavior. The two members are unable to verify the condition of their peer member router. Virtual Chassis Heartbeat uses the “no-split-detection” reactions, that requires VC-P to remain in the protocol master role and VC-B as VC-P. This “split-master” condition is not ideal for routing protocols and other topology management mechanisms. In this scenario, the split-master condition is better than operating without any protocol master member.

Virtual Chassis Heartbeat communication is active only when the Virtual Chassis is properly formed with successful election of VC-P and VC-B member chassis roles. Roles determined during a VCP adjacency loss are maintained until the Virtual Chassis is properly formed again. Disruption of Virtual Chassis Heartbeat connectivity does not impact protocol roles in the Virtual Chassis through the duration of "split" conditions.

Heartbeat Connection Compared to Split Detection

In certain Virtual Chassis failure conditions, the split detection setting (enabled by default, or explicitly disabled) can cause unpredictable and undesirable results such as a Virtual Chassis with two primary routers, or a Virtual Chassis with no primary router.

BEST PRACTICE: It is compulsory that you use the heartbeat connection instead of the split detection feature in an MX Series Virtual Chassis to avoid unnecessary primary role changes during an adjacency disruption or split, and to provide additional member health information for the primary-role election process.

Table 6 on page 24 compares the effects of split detection and the heartbeat connection for two common failure conditions: failure of the Virtual Chassis port interfaces and failure of the VC-B chassis.

Table 6: Comparison of Heartbeat Connection and Split Detection for Virtual Chassis Failure Conditions

Failure Condition	Results with Heartbeat Connection	Results with Split Detection
Virtual Chassis port interfaces go down.	• VC-P chassis retains VC-P role. • If the VC-P chassis is in service but the Virtual Chassis port interfaces are down, the VC-B chassis goes offline after the heartbeat timeout period expires because the Routing Engine state is invalid.	When split detection is disabled: • VC-P chassis retains VC-P role. • VC-B chassis also takes VC-P role. • Virtual Chassis has two primary routers, each of which maintains subscriber state information. 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.
VC-B chassis fails.	• VC-P chassis retains VC-P role. • VC-B chassis is out of service.	When split detection is enabled: • VC-P chassis takes line-card (VC-L) role, which isolates and removes it from the Virtual Chassis until connectivity is restored. • VC-B chassis is out of service. • Virtual Chassis does not have a primary router. This state halts interchassis routing and effectively disables the Virtual Chassis configuration.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you must configure an IP-based, bidirectional “heartbeat” packet connection between the primary router and backup router in an MX Series Virtual Chassis.

RELATED DOCUMENTATION

Primary-role Election in a Virtual Chassis

In a two-member MX Series or EX9200 *Virtual Chassis*, either member device can be elected as the primary device (also known as the protocol primary, or VC-P) of the Virtual Chassis. The first member device to join the Virtual Chassis becomes the initial primary device by default. After the Virtual Chassis is formed with both member devices, the Virtual Chassis Control Protocol (VCCP) software runs a primary-role election algorithm to elect the primary device for the Virtual Chassis configuration.

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

NOTE: You cannot configure primary-role election for an MX Series or EX9200 Virtual Chassis in the current release.

The VCCP software uses the following algorithm to elect the primary device for an EX9200 or MX Series Virtual Chassis:

1. Choose the member device that has the highest value for the internal primary-role election flag.

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

To display the primary-role election flag value, issue the `show virtual-chassis protocol database extensive` command. The flag value used for primary-role 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 device with the highest primary-role priority value.

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

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

   To display the primary-role priority value for the member devices in the Virtual Chassis, issue the `show virtual-chassis status` command.

3. Choose the member device that is active in the Virtual Chassis.

4. Choose the member device that belongs to the Virtual Chassis with the largest number of members.

   NOTE: This criterion is not used in the current release because all EX9200 and MX Series Virtual Chassis configurations have two member devices.

5. Choose the member device that is the accepted (elected) protocol primary of the Virtual Chassis.

6. Choose the member device that is the current protocol primary (VC-P) of the same Virtual Chassis.

7. Choose the member device that is the current protocol backup (VC-B) of the same Virtual Chassis.

8. Choose the member device that has been part of the Virtual Chassis configuration for the longest period of time.

9. Choose the member device that was the previous protocol primary of the same Virtual Chassis.

10. Choose the member device with the lowest media access control (MAC) address.

RELATED DOCUMENTATION

Virtual Chassis Components Overview | 7
Global Roles and Local Roles in a Virtual Chassis | 14
Switching the Global Primary and Backup Roles in a Virtual Chassis Configuration | 119

Switchover Behavior in an MX Series Virtual Chassis

IN THIS SECTION

When an active or primary hardware or software component fails or is temporarily shut down, you can manually initiate a *switchover* to a backup component that takes over the functions of the unavailable primary component. You can initiate two types of switchovers in a *Virtual Chassis* configuration for MX Series 5G Universal Routing Platforms:

- Global switchover—Changes the primary role in an MX Series Virtual Chassis by switching the global roles of the primary router and backup router in the Virtual Chassis configuration.
- Local switchover—Toggles the local primary role 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

To change the primary role in an MX Series Virtual Chassis and cause a global switchover, you issue the `request virtual-chassis routing-engine master switch` command from the primary Routing Engine in the Virtual Chassis primary router (VC-Pp).

After you issue the `request virtual-chassis routing-engine master switch` command, the current Virtual Chassis primary router (VC-P) and the current Virtual Chassis backup router (VC-B) switch roles. The former VC-P becomes the new VC-B, and the former VC-B becomes the new VC-P. After the VC-P and VC-B switch roles, the primary Routing Engine on the new VC-B (VC-Bp) reboots, causing the role transitions listed in Table 7 on page 28.

Table 7: Virtual Chassis Role Transitions During Global Switchover

Virtual Chassis Role *Before* Global Switchover	Virtual Chassis Role *After* Global Switchover
Virtual Chassis primary router (VC-P)	Virtual Chassis backup router (VC-B)
Virtual Chassis backup router (VC-B)	Virtual Chassis primary router (VC-P)
Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)
Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)	Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)
Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)	Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)
Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)

The local roles (`master` and `standby`, or `m` and `s`) of the Routing Engines in the Virtual Chassis primary router change after a global switchover, but the local roles of the Routing Engines in the Virtual Chassis backup router do not change. For example, as shown in Table 7 on page 28, the primary Routing Engine in the Virtual Chassis primary router (VC-Pp) becomes the standby Routing Engine in the Virtual Chassis backup router (VC-Bs) after the global switchover. By contrast, the primary Routing Engine in the Virtual Chassis backup router (VC-Bp) remains the primary Routing Engine in the Virtual Chassis primary router (VC-Pp) after the global switchover.

Virtual Chassis Role Transitions During a Local Switchover

To ensure redundancy in a two-member Virtual Chassis configuration, each of the two member routers must be configured with dual Routing Engines. To toggle local primary role between the primary Routing Engine and the standby Routing Engine in the member router, you issue the `request chassis routing-engine master switch` command from *either* the primary Routing Engine in the Virtual Chassis primary router (VC-Pp) or from the primary Routing Engine in the Virtual Chassis backup router (VC-Bp).

Table 8 on page 29 shows the role transitions caused by a local switchover when you issue the `request chassis routing-engine master switch` command from the VC-Pp.

Table 8: Virtual Chassis Role Transitions During Local Switchover Performed from VC-Pp

Virtual Chassis Role *Before* Local Switchover	Virtual Chassis Role *After* Local Switchover
Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)
Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)	Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)
Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)	Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)
Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)

Table 9 on page 29 shows the role transitions caused by a local switchover when you issue the `request chassis routing-engine master switch` command from the VC-Bp.

Table 9: Virtual Chassis Role Transitions During Local Switchover Performed from VC-Bp

Virtual Chassis Role *Before* Local Switchover	Virtual Chassis Role *After* Local Switchover
Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)
Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)

Table 9: Virtual Chassis Role Transitions During Local Switchover Performed from VC-Bp *(Continued)*

Virtual Chassis Role *Before* Local Switchover	Virtual Chassis Role *After* Local Switchover
Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)	Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)
Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)	Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)

When you perform a local switchover, the primary (m) and standby (s) local roles of the Routing Engines in each member router change only in the member router from which you issue the `request chassis routing-engine master switch` command. For example, when you issue a local switchover from the VC-Pp, as shown in Table 8 on page 29, the local roles change on the VC-P but remain the same on the VC-B. Conversely, when you issue a local switchover from the VC-Bp, as shown in Table 9 on page 29, the local roles change on the VC-B but remain the same on the VC-P.

A local switchover performed from the VC-Pp also changes the global roles of the member routers, as shown in Table 8 on page 29. By contrast, a local switchover performed from the VC-Bp changes only the local roles of the Routing Engines, as shown in Table 9 on page 29.

Virtual Chassis Role Transitions During Virtual Chassis Formation

In the rare case when the virtual chassis has "split" (that is, lost connectivity), each member may take the Virtual Chassis primary router (VC-P) role, resulting in two VC-P chassis. When Virtual Chassis connectivity is restored, an election process assigns the Virtual Chassis primary (VC-P) role to one member and the Virtual Chassis backup (VC-B) role to the other member. As of Junos OS Release 15.1, in the same manner as the global GRES behavior, the newly elected VC-B member causes its local primary Routing Engine to reboot after passing local primary role to its local standby Routing Engine. This is an intentional action which allows the VC-B chassis to become GRES-ready more quickly.

NOTE: Rebooting both Routing Engines in the VC-P chassis, or just the primary Routing Engine in either the VC-P or VC-B chassis, may not result in a graceful switchover and is not recommended.

Rebooting both Routing Engines in the VC-B chassis results in a VC split and there is no any RE role switchover.

GRES Readiness in a Virtual Chassis Configuration

Depending on the 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 primary router (VC-Pp) 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, completes before the global GRES operation begins.

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 or Virtual Chassis are synchronized and ready for the GRES operation.

RELATED DOCUMENTATION

Switching the Global Primary and Backup Roles in a Virtual Chassis Configuration | 119
Determining GRES Readiness in a Virtual Chassis Configuration | 241
Virtual Chassis Components Overview | 7
Global Roles and Local Roles in a Virtual Chassis | 14
Primary-role Election in a Virtual Chassis | 25
Understanding Graceful Routing Engine Switchover

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 10 on page 32 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, where *member-id* is 0 or 1.

Table 10: 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 member of the Virtual Chassis.	Change FPC status of the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Resynchronize the craft interface status on the specified member of the Virtual Chassis.
`request chassis pic`	Change the PIC status of the specified member router.	—	—	Change the PIC status on the specified member of the Virtual Chassis.
`request chassis routing-engine master`	Control which Routing Engine is the primary for a router with dual Routing Engines.	Control Routing Engine primary role on the Routing Engines in all member routers of the Virtual Chassis configuration.	(Default) Control Routing Engine primary role on the Routing Engines in the local Virtual Chassis configuration.	Control Routing Engine primary role on the Routing Engines of the specified member in the Virtual Chassis configuration.
`request chassis sfb`	(MX2010 and MX2020 routers only) Control the operation of the Switch Fabric Board (SFB).	Control the operation of the SFB in all members of the Virtual Chassis configuration.	(Default) Control the operation of the SFB in the local member of the Virtual Chassis.	Control the operation of the SFB in the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`request chassis spmb restart`	(MX2010 and MX2020 routers only) Restart the specified Switch Processor Mezzanine Board (SPMB) on the Control Board (CB).	Restart the SPMB on the CB in all members of the Virtual Chassis configuration.	(Default) Restart the SPMB on the CB in the local member of the Virtual Chassis.	Restart the SPMB on the CB in the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Log in to the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display system information for the specified member of the Virtual Chassis.
`request system halt`	Stop the router.	(Default) Halt all members of the Virtual Chassis configuration.	Halt the local member of the Virtual Chassis.	Halt the specified member of the Virtual Chassis.
`request system partition abort`	Terminate a previously scheduled storage media partition operation.	(Default) Terminate a previously scheduled storage media partition operation for all members of the Virtual Chassis configuration.	Terminate a previously scheduled storage media partition operation for the local member of the Virtual Chassis.	Terminate a previously scheduled storage media partition operation for the specified member of the Virtual Chassis member.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Schedule a partition of the hard disk for the specified member of the Virtual Chassis.
`request system power-off`	Power off the software.	(Default) Power off all members of the Virtual Chassis configuration.	Power off the local member of the Virtual Chassis.	Power off the specified member of the Virtual Chassis.
`request system reboot`	Reboot the software.	(Default) Reboot the software on all members of the Virtual Chassis configuration.	Reboot the software on the local member of the Virtual Chassis.	Reboot the software on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Archive data and executable areas for the specified member of the Virtual Chassis.
`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.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Attempt to roll back to the previous set of packages on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Validate the software bundle or package on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Delete files on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Restart the software process for a specified member of the Virtual Chassis.
`show chassis adc`	(MX2010 and MX2020 routers only) Display information about the adapter cards (ADCs).	(Default) Display information about the ADCs in all members of the Virtual Chassis configuration.	Display information about the ADCs in the local member of the Virtual Chassis.	Display information about the ADCs in the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display information about alarm conditions for the specified member of the Virtual Chassis.
`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 local member of the Virtual Chassis.	Display information currently on the craft interface for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display chassis environmental information for the specified member of the Virtual Chassis.
`show chassis environment adc`	(MX2010 and MX2020 routers only) Display chassis environmental information about the adapter cards (ADCs).	(Default) Display chassis environmental information about the ADCs in all members of the Virtual Chassis configuration.	Display chassis environmental information about the ADCs in the local member of the Virtual Chassis.	Display chassis environmental information about the ADCs in the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display environmental information about the CBs on the specified member of the Virtual Chassis.
`show chassis environment fan`	(MX2010 and MX2020 routers only) Display environmental information about the fans and fan trays.	(Default) Display environmental information about the fans and fan trays in all members of the Virtual Chassis configuration.	Display environmental information about the fans and fan trays in the local member of the Virtual Chassis.	Display environmental information about the fans and fan trays in the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display environmental information for the FPCs in the specified member of the Virtual Chassis.
`show chassis environment monitored`	(MX2010 and MX2020 routers only) Display status information for monitored temperatures.	(Default) Display status information for monitored temperatures in all members of the Virtual Chassis configuration.	Display status information for monitored temperatures in the local member of the Virtual Chassis.	Display status information for monitored temperatures in the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display environmental information about the PEMs in the specified member of the Virtual Chassis.
`show chassis environment psm`	(MX2010 and MX2020 routers only) Display chassis environmental information about the power supply modules (PSMs).	(Default) Display chassis environmental information about the PSMs in all member routers of the Virtual Chassis configuration.	Display chassis environmental information about the PSMs in the local member of the Virtual Chassis.	Display chassis environmental information about the PSMs in the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display environmental information about the Routing Engines in the specified member of the Virtual Chassis.
`show chassis environment sfb`	(MX2010 and MX2020 routers only) Display chassis environmental information about the Switch Fabric Boards (SFBs).	(Default) Display chassis environmental information about the SFBs in all member routers in the Virtual Chassis configuration.	Display chassis environmental information about the SFBs in the local member of the Virtual Chassis.	Display chassis environmental information about the SFBs in the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display information about the ports on the CB Ethernet switch on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display the switching fabric link states for the FPCs in the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display the switching fabric map state for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display the state of all fabric plane connections on the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`show chassis fabric plane-location`	Display the Control Board (CB) location of each plane on both the primary 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 member of the Virtual Chassis.	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 member of the Virtual Chassis.	Display information about the fan tray and fans for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display the version levels of the firmware running for the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display status information for all FPCs on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display hardware-specific information for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display the physical location of the chassis for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display the MAC addresses for the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display PIC information for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display power usage information for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display Routing Engine information for the specified member of the Virtual Chassis.
`show chassis sfb`	(MX2010 and MX2020 routers only) Display chassis information about the Switch Fabric Boards (SFBs).	(Default) Display chassis information about the SFBs in all members of the Virtual Chassis configuration.	Display chassis information about the SFBs in the local member of the Virtual Chassis.	Display chassis information about the SFBs in the specified member of the Virtual Chassis.
`show chassis spmb`	(MX2010 and MX2020 routers only) Display status information for the Switch Processor Mezzanine Boards (SPMBs).	(Default) Display status information for the SPMBs in all members of the Virtual Chassis configuration.	Display status information for the SPMBs in the local member of the Virtual Chassis.	Display status information for the SPMBs in the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display the chassis temperature threshold settings of the specified member of the Virtual Chassis.
`show chassis zones`	(MX2010 and MX2020 routers only) Display the status of the cooling system zones of the chassis.	(Default) Display the status of the cooling system zones in all members of the Virtual Chassis configuration.	Display the status of the cooling system zones in the local member of the Virtual Chassis.	Display the status of the cooling system zones in the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display Packet Forwarding Engine statistics for the specified FPC in the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display Packet Forwarding Engine status information for the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display file system MD5 hash and permissions information on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display boot time messages on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Show buffer statistics for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display system connection activity for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display directory information for the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display standard system process information for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display system queue statistics for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display halt or reboot request information for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display system statistics for a protocol for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display system storage statistics for the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display graceful Routing Engine switchover information for all Routing Engines on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Show time since the system rebooted and processes started on the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display users currently logged in to the specified member of the Virtual Chassis.

Table 10: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(Continued)*

Command	Purpose	`all-members`	`local`	`member` *`member-id`*
`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 member of the Virtual Chassis.	Display kernel dynamic memory usage information for the specified member of the Virtual Chassis.
`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 member of the Virtual Chassis.	Display standard information about the hostname and version of the software running on the specified member of the Virtual Chassis.
`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 primary and backup Routing Engines on all members of the Virtual Chassis configuration.	Display the hostname and version information about the software running on all primary and backup Routing Engines on the local member of the Virtual Chassis.	Display the hostname and version information about the software running on all primary and backup Routing Engines on the specified member of the Virtual Chassis.

RELATED DOCUMENTATION

Virtual Chassis Components Overview | 7

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

CLI Explorer

3

CHAPTER

Configuring a Virtual Chassis

IN THIS CHAPTER

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms 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.

NOTE: For recommended settings on MX Virtual Chassis devices, please consult Maximizing Scaling and Performance for MX Series Virtual Chassis on the Juniper Networks Knowledge Base.

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 51.

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.

3. Define configuration groups for the Virtual Chassis.

   See "Creating and Applying Configuration Groups for a Virtual Chassis" on page 54.

4. Create the preprovisioned member configuration on the primary router in the Virtual Chassis.

   See "Configuring Preprovisioned Member Information for a Virtual Chassis" on page 56.

5. Configure a heartbeat connection between the primary router and backup router.

   See "Configuring a Virtual Chassis Heartbeat Connection" on page 17

6. Configure enhanced IP network services on both member routers.

   See "Configuring Enhanced IP Network Services for a Virtual Chassis" on page 59.

7. 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 63.

8. Set the preprovisioned member IDs and reboot the routers in Virtual Chassis mode.

   See "Configuring Member IDs for a Virtual Chassis" on page 65.

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

   See "Configuring Virtual Chassis Ports to Interconnect Member Routers or Switches" on page 168.

10. (Optional) Starting in Junos OS Release 20.2R1, enable the ephemeral configuration database.

    See *Enabling and Configuring Instances of the Ephemeral Configuration Database*.

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

    See the following topics:

    - "Verifying the Status of Virtual Chassis Member Routers or Switches" on page 236
    - "Verifying the Operation of Virtual Chassis Ports" on page 237
    - "Verifying Neighbor Reachability for Member Routers or Switches in a Virtual Chassis" on page 238
    - "Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis" on page 240
    - "Viewing Information in the Virtual Chassis Control Protocol Adjacency Database" on page 242
    - "Viewing Information in the Virtual Chassis Control Protocol Link-State Database" on page 244
    - "Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database" on page 245
    - "Viewing Virtual Chassis Control Protocol Routing Tables" on page 247
    - "Viewing Virtual Chassis Control Protocol Statistics for Member Devices and Virtual Chassis Ports" on page 248

RELATED DOCUMENTATION

Interchassis Redundancy and Virtual Chassis Overview | **2**

Virtual Chassis Components Overview | **7**

Guidelines for Configuring Virtual Chassis Ports | **166**

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | **69**

Understanding the Ephemeral Configuration Database

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 primary or backup of the Virtual Chassis.

   > **NOTE**: When configuring multiple Routing Engines in a Virtual Chassis, all must have the same amount of physical memory allocated.

   - The *primary 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 primary router and relays chassis control information (such as line-card presence and alarms) to the primary router. If the primary router is unavailable, the backup router takes primary role 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.

   For the list of supported Routing Engines on MX series routers, see Supported Routing Engines by Router.

5. Ensure that the necessary 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 5G Universal Routing Platforms Using a Virtual Chassis | 50

Guidelines for Configuring Virtual Chassis Ports | 166

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Creating and Applying Configuration Groups for a Virtual Chassis

For a Virtual Chassis configuration consisting of two MX Series routers or two EX9200 switches, each of which supports dual Routing Engines, you must create and apply on the primary device 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 **member*n*-re*n*** naming format for configuration groups is reserved for exclusive use by member routers or switches in EX9200 or MX Series Virtual Chassis configurations.
>
> Using configuration group names of the form **member*n*-re*n*** in an existing non-Virtual Chassis configuration or configuration script could interfere with Virtual Chassis operation. This misconfiguration could cause the router or switch 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 or switch to be configured as the primary of the Virtual Chassis:

1. In the console window on the primary router or switch (**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@host# delete apply-groups re1
user@host# 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@host# set groups member1-re0 interfaces fxp0 unit unit-number family inet address address
user@host# 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@host# set groups member1-re1 system host-name host-name
user@host# set groups member1-re1 system backup-router address
user@host# set groups member1-re1 system backup-router destination destination-address
user@host# set groups member1-re1 system backup-router destination destination-address
...
```

```
user@host# set groups member1-re1 interfaces fxp0 unit unit-number family inet address address
user@host# 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 EX9200 or 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 a Virtual Chassis configuration has the same effect as issuing the `commit synchronize force` command.

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

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 primary 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 primary router or backup router of the Virtual Chassis.

3. (Optional) Enable locality bias in the Virtual Chassis configuration.

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

BEST PRACTICE: Starting in Junos OS Release 14.1, you can enable locality bias in the Virtual Chassis configuration. Locality bias can cause traffic loss and oversubscription on egress interfaces if you configure it in a network that is not designed to handle locality biasing. Make sure you understand the utilization requirements, such as total and available bandwidth, for the local links in your network before changing the locality bias configuration.

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;
locality-bias;
traceoptions {
    file vccp size 10m;
    flag all;
}
member 0 {
    role routing-engine;
    serial-number JN115FDADAFB;
}
member 1 {
    role routing-engine;
    serial-number JN10C78D1AFC;
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can enable locality bias in the Virtual Chassis configuration.

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring Enhanced IP Network Services for a Virtual Chassis

For an existing MX Series Virtual Chassis to function properly, you must configure enhanced IP network services on all member routers in the Virtual Chassis from the Virtual Chassis primary router.

Enhanced IP network services defines how the chassis recognizes and uses certain modules. When you set each member router's network services to `enhanced-ip`, only MPC/MIC modules and MS-DPC modules are powered on in the chassis. Non-service DPCs do not work with enhanced IP network services.

This procedure describes how to configure enhanced IP network services for an existing MX Series Virtual Chassis. For information about configuring enhanced IP network services when you first set up the Virtual Chassis, see *Configuring Enhanced IP Network Services* in "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69.

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.

To configure enhanced IP network services for an existing Virtual Chassis:

1. Log in to the console for the primary Routing Engine in the Virtual Chassis primary router (member0-re0 in this procedure).
2. Access the chassis hierarchy.

```
{master:member0-re0}[edit]
user@hostA# edit chassis
```

3. Configure enhanced IP network services on member 0.

```
{master:member0-re0}[edit chassis]
user@hostA# set network-services enhanced-ip
```

4. Commit the configuration.
5. When prompted to do so, reboot all Routing Engines in the Virtual Chassis.

```
{master:member0-re0}
user@hostA> request system reboot
```

The `request system reboot` command reboots both Routing Engines in each member router forming the Virtual Chassis.

6. (Optional) Verify that enhanced IP network services has been properly configured for the Virtual Chassis.

    a. Verify that enhanced IP network services is configured on the primary Routing Engine in the Virtual Chassis primary router (member0-re0).

    ```
    {master:member0-re0}
    user@hostA> show chassis network-services
    ```

    ```
    Network Services Mode: Enhanced-IP
    ```

    b. Verify that enhanced IP network services is configured on the primary Routing Engine in the Virtual Chassis backup router (member1-re0).

    ```
    {backup:member1-re0}
    user@hostB> show chassis network-services
    ```

    ```
    Network Services Mode: Enhanced-IP
    ```

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring Enhanced LAN Mode for a Virtual Chassis

Configuring `network-services lan` assumes the system is running in `network-services enhanced-ip` mode.

NOTE: Be sure to delete any unsupported configurations before changing to `enhanced-ip` mode.

To configure MX-LAN Mode for an existing Virtual Chassis:

1. Log into the console for the primary Routing Engine in the Virtual Chassis primary router (member0-re0 in this procedure).
2. Access the chassis hierarchy.

```
{master:member0-re0}[edit]
user@host# edit chassis
```

3. Configuring MX-LAN Mode on member 0.

```
{master:member0-re0}[edit chassis]
user@host# set network-services lan
```

4. Commit the configuration.
5. When prompted to do so, reboot all Routing Engines in the Virtual Chassis.

```
{master:member0-re0}
user@host> request system reboot
```

The `request system reboot` command reboots both Routing Engines in each member router forming the Virtual Chassis.

> **WARNING**: After the chassis configuration for network services has been changed, a system reboot is mandatory. Please reboot the system now. Continuing without a reboot might result in unexpected system behavior.

6. (Optional) Verify that enhanced IP network services has been properly configured for the Virtual Chassis.

   Verify that MX-LAN Mode is configured on the primary Routing Engine in the Virtual Chassis primary router (member0-re0).

```
{master:member0-re0}
user@host> show chassis network-services
```

```
Network Services Mode: MX-LAN
```

You must reboot the router when you configure or delete the enhanced LAN mode on the router. Configuring the `network-services lan` option implies that the system is running in the enhanced IP mode. When you configure a device to function in MX-LAN mode, only the supported configuration statements and operational show commands that are available for enabling or viewing in this mode are displayed in the CLI interface. If your system contains parameters that are not supported in MX-LAN mode in a configuration file, you cannot commit those unsupported attributes. You must remove the settings that are not supported and then commit the configuration. After the successful CLI commit, a system reboot is required for the attributes to be come effective. Similarly, if you remove the `network-services lan` statement, the system does not run in MX-LAN mode. Therefore, all of the settings that are supported outside of the MX-LAN mode are displayed and are available for definition in the CLI interface. If your configuration file contains settings that are supported only in MX-LAN mode, you must remove those attributes before you commit the configuration. After the successful CLI commit, a system reboot will be required for the CLI settings to take effect. The Layer 2 Next-Generation CLI configuration settings are supported in MX-LAN mode. As a result, the typical MX Series-format of CLI configurations might differ in MX-LAN mode.

For more information about the Layer 2 Next-Generation (L2NG) mode, also called Enhanced Layer 2 software (ELS), and the hierarchy levels at which the different configuration statements and commands are available for various parameters, see *Using the Enhanced Layer 2 Software CLI*.

RELATED DOCUMENTATION

Configuring Enhanced IP Network Services for a Virtual Chassis | 59

network-services

request system reboot

show chassis network-services

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. Configure the `commit` command to automatically result in a `commit synchronize` action between the dual Routing Engines in member 0.

   ```
   [edit system]
   user@gladius# set commit synchronize
   ```

   e. Commit the configuration.

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. Configure the `commit` command to automatically result in a `commit synchronize` action between the dual Routing Engines in member 1.

```
[edit system]
user@trefoil# set 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.

e. Commit the configuration.

NOTE: After enabling GRES, if you are using the ephemeral database, make sure to apply the `allow-commit-synchronize-with-gres` statement at the `[edit system configuration-database ephemeral]` hierarchy level. This enables the MX Series Virtual Chassis to synchronize ephemeral configuration data when a commit synchronize operation is requested.

NOTE: When you perform a GRES switchover while using enhanced subscriber management, the backup RE will reboot.

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring Graceful Routing Engine Switchover

Configuring Nonstop Active Routing

Enabling and Configuring Instances of the Ephemeral Configuration Database

Configuring Member IDs for a Virtual Chassis

After you commit the preprovisioned configuration on the primary 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. In an MX Series Virtual Chassis, you can optionally include the `slots-per-chassis` *`slot-count`* option to identify the number of chassis slots in the member router. Assigning the member ID and, optionally, *`slot-count`* 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.

For information about the supported member router combinations in an MX Series Virtual Chassis, see "Interchassis Redundancy and Virtual Chassis Overview" on page 2. Platform support depends on the Junos OS release in your installation.

To configure the member ID and, optionally, slot count for each member router in an MX Series Virtual Chassis:

1. Set the member ID on the router configured as member 0 in one of the following ways:

- For a Virtual Chassis consisting of a combination of MX240 routers, MX480 routers, and MX960 routers:

```
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
```

  With this Virtual Chassis configuration, the default *slot-count* (12) is the same for both member routers, so you do not need to configure it.

- For a Virtual Chassis consisting of an MX2010 router and either an MX960 router or MX2010 router:

```
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
```

  With this Virtual Chassis configuration, the default *slot-count* (12) is the same for an MX960 router and an MX2010 router, so you do not need to configure it.

- For a Virtual Chassis consisting of an MX2020 router and either an MX960 router or MX2010 router:

```
user@hostA> request virtual-chassis member-id set member 0 slots-per-chassis slot-count
```

```
This command will enable virtual-chassis mode and reboot the system.
 Continue? [yes,no] yes
```

  To ensure that this Virtual Chassis configuration forms properly, you *must* set the *slot-count* value of the MX960 router or MX2010 router to 20 to match the *slot-count* of the MX2020 router. For example:

```
user@hostA> request virtual-chassis member-id set member 0 slots-per-chassis 20
```

- For a Virtual Chassis consisting of two MX2020 routers:

```
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
```

  With this Virtual Chassis configuration, the default *slot-count* (20) is the same for both MX2020 routers, so you do not need to configure it.

The router reboots in preparation for forming the Virtual Chassis. After the reboot, all MPCs remain powered off until the Virtual Chassis port connection is configured.

2. Repeat Step 1 to 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
```

or

```
user@hostB> request virtual-chassis member-id set member 1 slots-per-chassis slot-count
```

```
This command will enable virtual-chassis mode and reboot the system.
 Continue? [yes,no] yes
```

For example:

```
user@hostA> request virtual-chassis member-id set member 0 slots-per-chassis 20
```

The router reboots in preparation for forming the Virtual Chassis. After the reboot, all MPCs remain powered off until the Virtual Chassis port connection is configured.

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 5G Universal Routing Platforms Using a Virtual Chassis | 50

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis

IN THIS SECTION

To provide interchassis redundancy for MX Series 5G Universal Routing Platforms, 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 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

This example uses the following software and hardware components:

- Junos OS Release 11.2 and later releases

- One MX240 Universal Routing Platform
- One MX480 Universal Routing Platform

> **NOTE**: This configuration example has been tested using the software release listed and is assumed to work on all later releases.

See Table 11 on page 72 for information about the hardware installed in each MX Series router.

> **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.

Overview and Topology

IN THIS SECTION

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 primary 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 primary 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.

Topology

This example configures a Virtual Chassis that interconnects two MX Series routers, and uses the basic topology shown in Figure 2 on page 71. 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

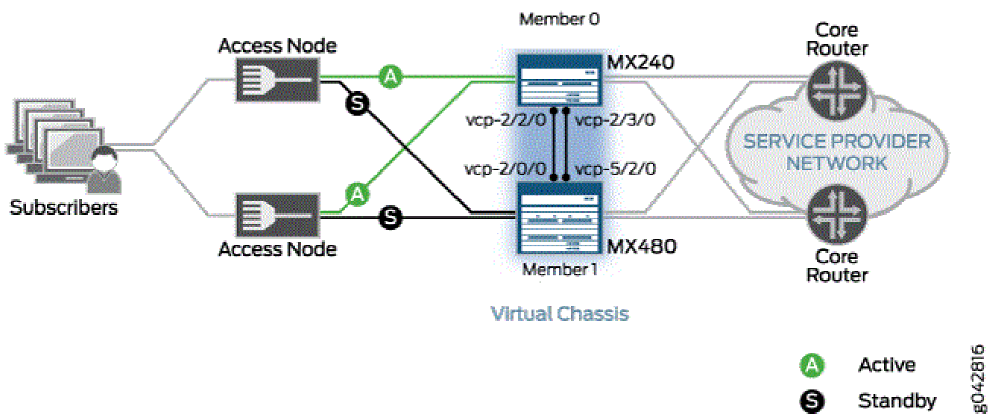


Table 11 on page 72 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis. You use some of these settings in the preprovisioned configuration and when you assign the member IDs and create the Virtual Chassis ports.

> **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.

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: • 60-Gigabit Ethernet Enhanced Queuing MPC • 20-port Gigabit Ethernet MIC with SFP • 4-port 10-Gigabit Ethernet MIC with XFP • Primary 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 (primary)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 11: 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: • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Primary 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

IN THIS SECTION

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

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                                JN10C7135AFC      MX240
.
.
.
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 primary or backup of the Virtual Chassis.

   - The *primary 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 primary router and relays chassis control information (such as line-card presence and alarms) to the primary router. If the primary router is unavailable, the backup router takes primary role 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 primary router is assigned member ID 0, and the backup router is assigned member ID 1.

4. Configure a heartbeat connection between the primary router and backup router.

   Starting from Junos release version 14.1R1, heartbeat connection is required on a virtual chassis configuration, to avoid unnecessary primary role changes during an adjacency disruption or split. Additionally, to add member health information for the primary-role election process.

5. 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.

6. Ensure that the necessary 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.

7. 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.

8. 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.

9. 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.

10. Install the MX Virtual Chassis Redundancy Feature Pack license on each router to be configured as part of the Virtual Chassis.

11. 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 primary of the Virtual Chassis instead of using the standard re0 and re1 configuration groups:

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

> **NOTE**: The `member`*n*`-re`*n* 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.

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. (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
```

5. Commit the configuration.

Results

Display the results of the configuration.

```
[edit virtual-chassis]
user@gladius# show
preprovisioned;
traceoptions {
    file vccp size 100m;
    flag all;
}
member 0 {
    role routing-engine;
    serial-number JN10C7135AFC;
}
member 1 {
    role routing-engine;
    serial-number JN115D117AFB;
}
```

Configuring a Virtual Chassis Heartbeat Connection

Starting in Junos OS Release 14.1, you must configure an IP-based, bidirectional “heartbeat” packet connection between the primary router and backup router in a virtual chassis. The heartbeat connection determines the health and availability of member routers in the virtual chassis. The member routers forming this heartbeat connection exchange *heartbeat packets* that provide critical information about the availability and health of each member router.

For details, see Configuring a Virtual Chassis Heartbeat Connection.

Configuring Enhanced IP Network Services

Step-by-Step Procedure

For an MX Series Virtual Chassis to function properly, you must configure enhanced IP network services on both member routers (member 0 and member 1). Enhanced IP network services defines how the chassis recognizes and uses certain modules. When you set each member router's network services to `enhanced-ip`, only MPC/MIC modules and MS-DPC modules are powered on in the chassis. Non-service DPCs do not work with enhanced IP network services.

This procedure describes how to configure enhanced IP network services when you first set up the Virtual Chassis. For information about configuring enhanced IP network services for an existing MX Series Virtual Chassis, see "Configuring Enhanced IP Network Services for a Virtual Chassis" on page 59.

To configure enhanced IP network services for a Virtual Chassis:

1. Configure enhanced IP network services on member 0 (`gladius`).

   a. Log in to the console on member 0.

   b. Access the chassis hierarchy.

   ```
   [edit]
   user@gladius# edit chassis
   ```

   c. Configure enhanced IP network services for member 0.

   ```
   [edit chassis]
   user@gladius# set network-services enhanced-ip
   ```

   d. Commit the configuration on member 0.

   NOTE: Immediately after you commit the configuration, the software prompts you to reboot the router. You can proceed without rebooting the router at this point because a reboot occurs when you configure the member IDs to enable Virtual Chassis mode, later in this procedure.

2. Configure enhanced IP network services on member 1 (`trefoil`).

   a. Log in to the console on member 1.

b. Access the chassis hierarchy.

```
[edit]
user@trefoil# edit chassis
```

c. Configure enhanced IP network services for member 1.

```
[edit chassis]
user@trefoil# set network-services enhanced-ip
```

d. Commit the configuration on member 1.

> **NOTE**: Immediately after you commit the configuration, the software prompts you to reboot the router. You can proceed without rebooting the router at this point because a reboot occurs when you configure the member IDs to enable Virtual Chassis mode, later in this procedure.

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. Configure the `commit` command to automatically result in a `commit synchronize` action between the dual Routing Engines in member 0.

```
[edit system]
user@gladius# set commit synchronize
```

e. Commit the configuration.

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. Configure the `commit` command to automatically result in a `commit synchronize` action between the dual Routing Engines in member 1.

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

e. Commit the configuration.

> **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.

After the reboot, all MPCs remain powered off until the Virtual Chassis port connection is configured.

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
```

After the reboot, all MPCs remain powered off until the Virtual Chassis port connection is configured.

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 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 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 primary 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 primary router.

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

Verification

IN THIS SECTION

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

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

                                               Mastership          Neighbor List
Member ID       Status    Serial No    Model    priority    Role      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 primary 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 Enhanced IP Network Services Configuration

Purpose

Verify that enhanced IP network services has been properly configured for the Virtual Chassis.

Action

Display the setting of the network services configuration for the primary Routing Engine in the Virtual Chassis primary router (member0-re0), and for the primary Routing Engine in the Virtual Chassis backup router (member1-re0).

```
{master:member0-re0}
user@gladius> show chassis network-services
Network Services Mode: Enhanced-IP
```

```
{backup:member1-re0}
user@trefoil> show chassis network-services
Network Services Mode: Enhanced-IP
```

Meaning

The output of the `show chassis network services` command confirms that enhanced IP network services is properly configured on both member routers in the Virtual Chassis.

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

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis

IN THIS SECTION

To provide interchassis redundancy for MX Series 5G Universal Routing Platforms, 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 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

This example uses the following software and hardware components:

- Junos OS Release 11.2 and later releases
- One MX240 Universal Routing Platform
- One MX480 Universal Routing Platform

NOTE: This configuration example has been tested using the software release listed and is assumed to work on all later releases.

See Table 11 on page 72 for information about the hardware installed in each MX Series router.

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.

Overview and Topology

IN THIS SECTION

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 primary 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 primary 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.

Topology

This example configures a Virtual Chassis that interconnects two MX Series routers, and uses the basic topology shown in Figure 2 on page 71. For redundancy, two Virtual Chassis ports are configured on each member router.

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

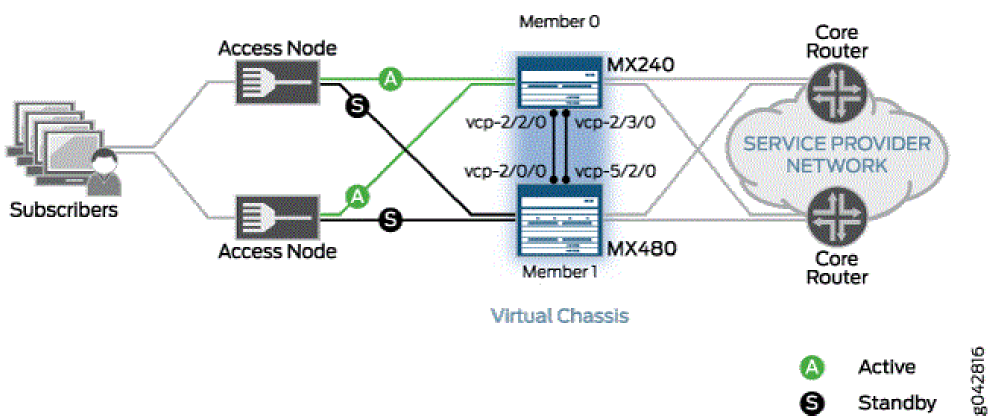


Table 11 on page 72 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis. You use some of these settings in the preprovisioned configuration and when you assign the member IDs and create the Virtual Chassis ports.

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.

Table 12: 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: • 60-Gigabit Ethernet Enhanced Queuing MPC • 20-port Gigabit Ethernet MIC with SFP • 4-port 10-Gigabit Ethernet MIC with XFP • Primary 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 (primary)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 12: 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: • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Primary 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

IN THIS SECTION

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

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                                JN10C7135AFC      MX240
.
.
.
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 primary or backup of the Virtual Chassis.

   - The *primary 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 primary router and relays chassis control information (such as line-card presence and alarms) to the primary router. If the primary router is unavailable, the backup router takes primary role 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 primary router is assigned member ID 0, and the backup router is assigned member ID 1.

4. Configure a heartbeat connection between the primary router and backup router.

   Starting from Junos release version 14.1R1, heartbeat connection is required on a virtual chassis configuration, to avoid unnecessary primary role changes during an adjacency disruption or split. Additionally, to add member health information for the primary-role election process.

5. 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.

6. Ensure that the necessary 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.

7. 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.

8. 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.

9. 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.

10. Install the MX Virtual Chassis Redundancy Feature Pack license on each router to be configured as part of the Virtual Chassis.

11. 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 primary of the Virtual Chassis instead of using the standard re0 and re1 configuration groups:

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

> **NOTE**: The `member`*n*`-re`*n* 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.

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. (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
```

5. Commit the configuration.

Results

Display the results of the configuration.

```
[edit virtual-chassis]
user@gladius# show
preprovisioned;
traceoptions {
    file vccp size 100m;
    flag all;
}
member 0 {
    role routing-engine;
    serial-number JN10C7135AFC;
}
member 1 {
    role routing-engine;
    serial-number JN115D117AFB;
}
```

Configuring a Virtual Chassis Heartbeat Connection

Starting in Junos OS Release 14.1, you must configure an IP-based, bidirectional "heartbeat" packet connection between the primary router and backup router in a virtual chassis. The heartbeat connection determines the health and availability of member routers in the virtual chassis. The member routers forming this heartbeat connection exchange *heartbeat packets* that provide critical information about the availability and health of each member router.

For details, see Configuring a Virtual Chassis Heartbeat Connection.

Configuring Enhanced IP Network Services

Step-by-Step Procedure

For an MX Series Virtual Chassis to function properly, you must configure enhanced IP network services on both member routers (member 0 and member 1). Enhanced IP network services defines how the chassis recognizes and uses certain modules. When you set each member router's network services to `enhanced-ip`, only MPC/MIC modules and MS-DPC modules are powered on in the chassis. Non-service DPCs do not work with enhanced IP network services.

This procedure describes how to configure enhanced IP network services when you first set up the Virtual Chassis. For information about configuring enhanced IP network services for an existing MX Series Virtual Chassis, see "Configuring Enhanced IP Network Services for a Virtual Chassis" on page 59.

To configure enhanced IP network services for a Virtual Chassis:

1. Configure enhanced IP network services on member 0 (`gladius`).

   a. Log in to the console on member 0.

   b. Access the chassis hierarchy.

   ```
   [edit]
   user@gladius# edit chassis
   ```

   c. Configure enhanced IP network services for member 0.

   ```
   [edit chassis]
   user@gladius# set network-services enhanced-ip
   ```

   d. Commit the configuration on member 0.

   NOTE: Immediately after you commit the configuration, the software prompts you to reboot the router. You can proceed without rebooting the router at this point because a reboot occurs when you configure the member IDs to enable Virtual Chassis mode, later in this procedure.

2. Configure enhanced IP network services on member 1 (`trefoil`).

   a. Log in to the console on member 1.

b. Access the chassis hierarchy.

```
[edit]
user@trefoil# edit chassis
```

c. Configure enhanced IP network services for member 1.

```
[edit chassis]
user@trefoil# set network-services enhanced-ip
```

d. Commit the configuration on member 1.

> **NOTE**: Immediately after you commit the configuration, the software prompts you to reboot the router. You can proceed without rebooting the router at this point because a reboot occurs when you configure the member IDs to enable Virtual Chassis mode, later in this procedure.

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. Configure the `commit` command to automatically result in a `commit synchronize` action between the dual Routing Engines in member 0.

```
[edit system]
user@gladius# set commit synchronize
```

e. Commit the configuration.

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. Configure the `commit` command to automatically result in a `commit synchronize` action between the dual Routing Engines in member 1.

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

e. Commit the configuration.

> **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.

After the reboot, all MPCs remain powered off until the Virtual Chassis port connection is configured.

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
```

After the reboot, all MPCs remain powered off until the Virtual Chassis port connection is configured.

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 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 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 primary 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 primary router.

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

Verification

IN THIS SECTION

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

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

                                              Mastership          Neighbor List
Member ID       Status   Serial No    Model    priority    Role      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 primary 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 Enhanced IP Network Services Configuration

Purpose

Verify that enhanced IP network services has been properly configured for the Virtual Chassis.

Action

Display the setting of the network services configuration for the primary Routing Engine in the Virtual Chassis primary router (member0-re0), and for the primary Routing Engine in the Virtual Chassis backup router (member1-re0).

```
{master:member0-re0}
user@gladius> show chassis network-services
Network Services Mode: Enhanced-IP
```

```
{backup:member1-re0}
user@trefoil> show chassis network-services
Network Services Mode: Enhanced-IP
```

Meaning

The output of the `show chassis network services` command confirms that enhanced IP network services is properly configured on both member routers in the Virtual Chassis.

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

RELATED DOCUMENTATION

Interchassis Redundancy and Virtual Chassis Overview | 2

Virtual Chassis Components Overview | 7

Guidelines for Configuring Virtual Chassis Ports | 166

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

RELATED DOCUMENTATION

Interchassis Redundancy and Virtual Chassis Overview | 2

Virtual Chassis Components Overview | 7

Guidelines for Configuring Virtual Chassis Ports | 166

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an MX2020 Member Router in an Existing MX Series Virtual Chassis

Starting in Junos OS Release 15.1, in an existing two-member MX Series Virtual Chassis that includes an MX960 router or an MX2010 router, you can replace one or both of these routers with an MX2020 router. If you replace either the MX960 router or MX2010 router with an MX2020 router, you must follow this procedure to ensure that the new Virtual Chassis forms properly and that any configurations you use reflect the correct Flexible PIC Concentrator (FPC) slot numbering for interfaces configured on the MX2020 member router.

Before you begin:

- Ensure that the existing Virtual Chassis is properly configured and operational before replacing one of the member routers with an MX2020 router.

  See "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69.

- Make sure you know the member router combinations supported with an MX2020 router.

  See "Interchassis Redundancy and Virtual Chassis Overview" on page 2.

- Make sure you understand the slot count values for MX960, MX2010, and MX2020 member routers, and how the FPC slots are numbered when you configure a Virtual Chassis with an MX2020 router.

  See "Virtual Chassis Components Overview" on page 7.

To configure an MX2020 member router in an existing MX Series Virtual Chassis with either an MX960 router or an MX2010 router:

1. Replace either the MX960 router or MX2010 router with an MX2020 router.

   Install the interface modules in the same physical slots in the MX2020 router as they were when installed in the MX960 or MX2010 router.
2. Upgrade all four Routing Engines in the Virtual Chassis (two Routing Engines in each member router) to the current Junos OS software release.

   > **NOTE**: All four Routing Engines in the Virtual Chassis must be the same model and running the same Junos OS software release. For a Virtual Chassis that includes an MX2020 router, all four Routing Engines must have at least 16 gigabytes of memory.
3. Set the member ID and slot count of the MX960 router or MX2010 router.

To ensure that a Virtual Chassis configuration with an MX2020 router and either an MX960 router or MX2010 router forms properly, you *must* set the *slot-count* value of the MX960 router or MX2010 router to 20 to match the *slot-count* of the MX2020 router.

```
user@hostA> request virtual-chassis member-id set member member-id slots-per-chassis slot-
count
```

For example, assume that the MX960 router or MX2010 router is member 1 in the preprovisioned Virtual Chassis configuration. To set the member ID and slot count for member 1:

```
user@hostA> request virtual-chassis member-id set member 1 slots-per-chassis 20
```

```
This command will enable virtual-chassis mode and reboot the system.
 Continue? [yes,no] yes
```

The router reboots in preparation for forming the Virtual Chassis. After the reboot, the FPC slots are renumbered and all MPCs remain powered off until the Virtual Chassis port connection is configured.

4. Edit your Junos OS configuration to rename any interfaces configured on member 1 to reflect how the FPC slots are renumbered when the Virtual Chassis includes an MX2020 member router.

   Before you add the MX2020 router to the Virtual Chassis, member 0 uses FPC slot numbers 0 through 11 with no offset, and member 1 uses slot numbers 12 through 23 with an offset of 20. After you set the slot count of the MX960 or MX2010 router to 20 and reboot the MX2020 member router, member 0 uses slot numbers 0 through 19 with no offset, and member 1 uses slot numbers 20 through 39 with an offset of 20.

   For example, in the following partial configuration, ge-0/0/0 is an aggregated Ethernet link configured on FPC slot 0 in member 0, and ge-12/0/0 is an aggregated Ethernet link configured on FPC slot 0 in member 1.

```
[edit]
interfaces {
    ge-0/0/0 {
        gigether-options {
            802.3ad ae0;
        }
    }
    ge-12/0/0 {
        gigether-options {
            802.3ad ae0;
```

```
        }
    }
    ae0 {
        ...
        ...
    }
}
```

If you replace member 1 with an MX2020 member router, FPC slot 0 in member 1 is renumbered as FPC slot 20 on member 1. As a result, you must edit the configuration to change ge-12/0/0 to ge-20/0/0, as follows:

```
[edit]
interfaces {
    ge-0/0/0 {
        gigether-options {
            802.3ad ae0;
        }
    }
    ge-20/0/0 {
        gigether-options {
            802.3ad ae0;
        }
    }
    ae0 {
        ...
        ...
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
15.1	Starting in Junos OS Release 15.1, in an existing two-member MX Series Virtual Chassis that includes an MX960 router or an MX2010 router, you can replace one or both of these routers with an MX2020 router.

RELATED DOCUMENTATION

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

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

Before you begin:

- Make sure the system configuration is synchronized between the primary 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 primary and backup roles:

- Issue the `request virtual-chassis routing-engine master switch` command from the Virtual Chassis primary Routing Engine in the Virtual Chassis primary router (VC-Pp):

```
{master:member0-re0}
user@host> 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 VC-Pp causes the global role transitions listed in Table 13 on page 120.

Table 13: Virtual Chassis Global Role Transitions Before and After Primary-role Switchover

Virtual Chassis Role Before Switching Primary Role	Virtual Chassis Role After Switching Primary Role
Primary Routing Engine in Virtual Chassis primary router (VC-Pp)	Standby Routing Engine in Virtual Chassis backup router (VC-Bs)
Standby Routing Engine in Virtual Chassis primary router (VC-Ps)	Primary Routing Engine in Virtual Chassis backup router (VC-Bp)
Primary Routing Engine in Virtual Chassis backup router (VC-Bp)	Primary Routing Engine in Virtual Chassis primary router (VC-Pp)
Standby Routing Engine in Virtual Chassis backup router (VC-Bs)	Standby Routing Engine in Virtual Chassis primary router (VC-Ps)

RELATED DOCUMENTATION

Switchover Behavior in an MX Series Virtual Chassis | 27
Virtual Chassis Components Overview | 7
Global Roles and Local Roles in a Virtual Chassis | 14
Primary-role Election in a Virtual Chassis | 25

Deleting Member IDs in a Virtual Chassis Configuration

In most cases, you delete the member ID from a member router or switch 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 or switch reboots and the software disables Virtual Chassis mode on that device. After the reboot, the router or switch is no longer part of the Virtual Chassis and functions as an independent device.

> **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.
>
> A software license is not needed to create an EX9200 Virtual Chassis.

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

1. In the console window on the router or switch 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 or switch 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 or switches.

   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 5G Universal Routing Platforms | **137**
Configuring an EX9200 Virtual Chassis
Example: Deleting a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms | **138**

Example: Replacing a Routing Engine in a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms

IN THIS SECTION

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 primary 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

This example uses the following software and hardware components:

- Junos OS Release 11.4 and later releases
- One MX240 Universal Routing Platform with dual Routing Engines
- One MX480 Universal Routing Platform with dual Routing Engines

NOTE: This configuration example has been tested using the software release listed and is assumed to work on all later releases.

See Table 14 on page 126 for information about the hardware installed in each MX Series router.

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.

Overview and Topology

IN THIS SECTION

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. Modify the Routing Engine factory configuration to enable formation of the Virtual Chassis.
5. Install the same Junos OS release on the new Routing Engine that is running on the other Routing Engines in the Virtual Chassis.
6. Reboot the new Routing Engine to run the Junos OS software release.

Topology

Figure 4 on page 125 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 4: Sample Topology for a Virtual Chassis with Two MX Series Routers

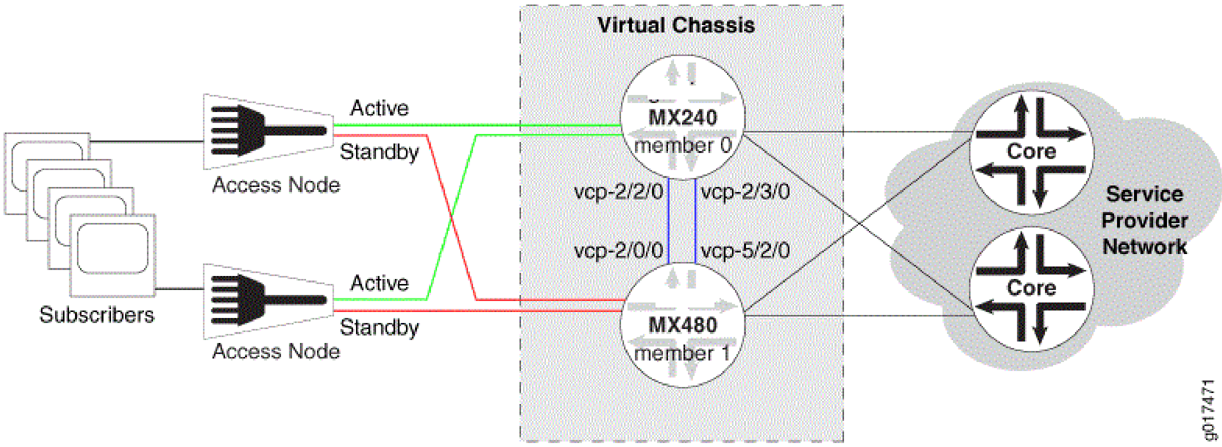


Table 14 on page 126 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 14: 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: • 60-Gigabit Ethernet Enhanced Queuing MPC • 20-port Gigabit Ethernet MIC with SFP • 4-port 10-Gigabit Ethernet MIC with XFP • Primary 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 (primary)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 14: 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: • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Primary 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

IN THIS SECTION

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

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 *How to Return 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*.

Modifying the Routing Engine Factory Configuration

Step-by-Step Procedure

A Routing Engine shipped from the factory is loaded with a default factory configuration that includes the following stanza at the [edit] hierarchy level:

```
[edit]
system {
    commit {
        factory-settings {
            reset-virtual-chassis-configuration;
        }
    }
}
```

When this configuration stanza is present, the Routing Engine can operate only in a standalone chassis and *not* in a Virtual Chassis member router. As a result, if you install this Routing Engine in the standby slot of a Virtual Chassis member router (`member1-re1` in this procedure), the Routing Engine does not automatically synchronize with the primary Routing Engine and boot in Virtual Chassis mode.

To ensure that the standby factory Routing Engine successfully synchronizes with the primary Routing Engine, you must remove the standalone chassis configuration stanza from the standby factory Routing Engine and verify that it reboots in Virtual Chassis mode before you install the Junos OS release.

To modify the Routing Engine factory configuration to ensure proper operation of the Virtual Chassis:

1. Log in to the console of the new Routing Engine as the user `root` with no password.

2. 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
```

3. Delete the standalone chassis configuration.

```
{local:member1-re1}[edit]
root# delete system commit factory-settings reset-virtual-chassis-configuration
```

4. Commit the configuration.

   The new Routing Engine synchronizes the Virtual Chassis member ID with the primary Routing Engine and boots in Virtual Chassis mode.

5. Verify that the new Routing Engine is in Virtual Chassis mode.

   During the boot process, the router displays the following output to indicate that it has synchronized the Virtual Chassis member ID (1) with the primary Routing Engine and is in Virtual Chassis mode.

```
...
virtual chassis member-id = 1
virtual chassis mode      = 1
...
```

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 primary Routing Engine on the Virtual Chassis primary router (`member0-re0`).

   See *Downloading Software* in the Junos OS Software 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. If you have not already done so, log in to the console of the new Routing Engine as the user `root` with no password.

3. If you have not already done so, 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. Log in to the console of the Virtual Chassis primary router (`member0-re0`) as the user `root`.

5. From the console of the Virtual Chassis primary router, commit the configuration.

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

6. 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.

7. Install the Junos OS release on the new Routing Engine (`member1-re1`) from the Virtual Chassis primary 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

IN THIS SECTION

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

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
- 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 Primary Role 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 primary role from the existing Routing Engine in the Virtual Chassis backup router, `trefoil` (member 1).

Action

Switch the local primary role 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

                                              Mastership           Neighbor List
Member ID         Status    Serial No    Model    priority    Role       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 primary role 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 primary router, and member 1 (`trefoil`) is still the Virtual Chassis backup router.

Before the local switchover, `member1-re0` was the primary Routing Engine in the Virtual Chassis backup router (VC-Bp), 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 primary Routing Engine in the Virtual Chassis backup router (VC-Bp), and member1-re0 becomes the standby Routing Engine in the Virtual Chassis backup router (VC-Bs).

Table 15 on page 136 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 15 on page 136 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 primary router (VC-P), see "Switchover Behavior in an MX Series Virtual Chassis" on page 27.

Table 15: Virtual Chassis Role Transitions Before and After Local Routing Engine Switchover

Virtual Chassis Component	Role *Before* Local Switchover	Role *After* Local Switchover
gladius (member 0)	Virtual Chassis primary router (VC-P)	Virtual Chassis primary router (VC-P)
trefoil (member 1)	Virtual Chassis backup router (VC-B)	Virtual Chassis backup router (VC-B)
member0-re0	Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)	Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)
member0-re1	Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)	Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)
member1-re0	Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)	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)	Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)

BEST PRACTICE: After you switch the local primary role 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 | 2
Virtual Chassis Components Overview | 7
Installing Junos OS Licenses on Virtual Chassis Member Routers
Switchover Behavior in an MX Series Virtual Chassis | 27
Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69
Junos OS Software Installation and Upgrade Guide

Deleting a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms

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 171.

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 primary 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 121.

> **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 5G Universal Routing Platforms

IN THIS SECTION

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

This example uses the following software and hardware components:

- Junos OS Release 11.2 and later releases
- One MX240 Universal Routing Platform with dual Routing Engines
- One MX480 Universal Routing Platform with dual Routing Engines

> **NOTE**: This configuration example has been tested using the software release listed and is assumed to work on all later releases.

SeeTable 16 on page 142 for information about the hardware installed in each MX Series router.

> **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.

Overview and Topology

IN THIS SECTION

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 primary 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 primary router. For example, even though the preprovisioned member information was configured at the `[edit virtual-chassis]` hierarchy level only on the primary router, you must delete the `virtual-chassis` stanza from the other member router in the Virtual Chassis.

NOTE: When deleting the Virtual Chassis, you must also delete all Virtual Chassis-related configuration details from all stanzas, otherwise errors will result upon commit.

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.

Topology

This example deletes the Virtual Chassis configuration that uses the basic topology shown in Figure 5 on page 141. 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 5: Sample Topology for a Virtual Chassis with Two MX Series Routers

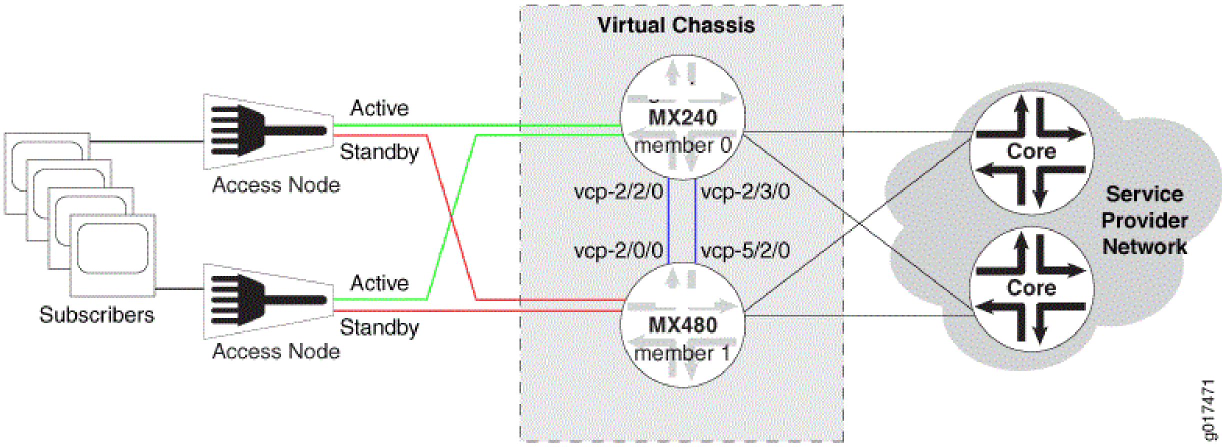


Table 16 on page 142 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 16: 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: • 60-Gigabit Ethernet Enhanced Queuing MPC • 20-port Gigabit Ethernet MIC with SFP • 4-port 10-Gigabit Ethernet MIC with XFP • Primary 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 (primary)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 16: 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: • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Primary 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

IN THIS SECTION

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

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 `member`*n*`-re`*n* 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]` hierarachy level on the primary 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
```

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

IN THIS SECTION

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

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 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`, `member 0-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

Upgrading an MX Virtual Chassis SCB or SCBE to SCBE2

IN THIS SECTION

To upgrade an MX Virtual Chassis SCB or SCBE to SCBE2, perform the following steps:

NOTE: SCBE2 does not support smooth upgrade.

Preparing for the SCBE2 Upgrade

To prepare for the SCBE2 upgrade:

1. Verify that the system is runing Junos OS Release 13.3 or later by issuing the **show version** command on the primary router.

```
user@host> show version
Junos Base OS Software Suite [13.3-yyyymmdd];
...
```

> **NOTE**: The SCBE2 is supported only on:
>
> - Junos OS Release 13.3 or later
> - Network Services Mode: Enhanced-IP

   The latest software ensures a healthy system—that is, a system that comprises Routing Engines, control boards, and FPCs—before the upgrade.

   For information about how to verify and upgrade Junos OS on MX Virtual Chassis configurations, see "Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms by Rebooting the Routing Engines" on page 209.

2. Record the local IP address of each RE in the Virtual Chassis configuration. This information will be necessary to reconfigure the VC member IDs after the upgrade.
3. Apply a system halt before shutting down the system by issuing the **request system halt all-members** command on the primary router.

```
user@host> request system halt all-members
```

Powering Off the MX Series Router

> **NOTE**: After turning off the power supply, wait at least 60 seconds before turning it back on.

To power off the MX Series router:

1. On the external management device connected to the Routing Engine, issue the `request system halt all-members` operational mode command. The command shuts down the Routing Engines cleanly, so that their state information is preserved.

```
user@host> request system halt all-members
```

2. Wait until a message appears on the console confirming that the operating system has halted.
3. Attach an electrostatic discharge (ESD) grounding strap to your bare wrist and connect the strap to one of the ESD points on the chassis.
4. Move the AC input switch on the chassis above the AC power supply or the DC circuit breaker on each DC power supply faceplate to the off **(O)** position.

Removing an MX Series Routing Engine from an SCB or SCBE

To remove an MX Series Routing Engine from an SCB or SCBE:

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*.

Replacing the SCB or SCBE with SCBE2

To replace the existing SCB or SCBE with SCBE2:

1. Attach an electrostatic discharge (ESD) grounding strap to your bare wrist and connect the strap to one of the ESD points on the chassis.
2. Remove and replace the offline SCB or SCBE on the router with SCBE2.

Installing the MX Series Routing Engine into an SCBE2

After removing the routing engine from the original SCB or SCBE, you can install it into the new SCBE2. Follow these instructions to install a MX Series Routing Engine into an SCBE2:

1. Attach an electrostatic discharge (ESD) grounding strap to your bare wrist, and connect the strap to one of the ESD points on the chassis.
2. Ensure that the ejector handles are not in the locked position. If necessary, flip the ejector handles outward.
3. Place one hand underneath the Routing Engine to support it.
4. Carefully align the sides of the Routing Engine with the guides inside the opening on the SCBE2.
5. Slide the Routing Engine into the SCBE2 until you feel resistance and then press the faceplate of the Routing Engine until it engages the connectors.
6. Press both of the ejector handles inward to seat the Routing Engine.
7. Tighten the captive screws on the top and bottom of the Routing Engine.
8. Connect the management device cables to the Routing Engine.

Powering On the MX Series Router

To power on the MX Series router:

1. Verify that the power supplies are fully inserted in the chassis.
2. Verify that each AC power cord is securely inserted into its appliance inlet.
3. Verify that an external management device is connected to one of the Routing Engine ports (**AUX**, **CONSOLE**, or **ETHERNET**).
4. Turn on the power to the external management device.
5. Switch on the dedicated customer-site circuit breakers. Follow the ESD and safety instructions for your site.
6. Attach an ESD grounding strap to your bare wrist and connect the strap to one of the ESD points on the chassis.
7. Move the AC input switch on the chassis above the AC power supply or the DC circuit breaker on each DC power-supply faceplate to the off (**—**) position.
8. Check that the AC or the DC power supply is correctly installed and functioning normally. Verify that the **AC OK** and **DC OK** LEDs light steadily, and the **PS FAIL** LED is not lit.

NOTE: After a power supply is powered on, it can take up to 60 seconds for status indicators—such as the status LEDs on the power supply and the `show chassis` command display—to indicate that the power supply is functioning normally. Ignore error indicators that appear during the first 60 seconds.

If any of the status LEDs indicates that the power supply is not functioning normally, repeat the installation and cabling procedures.

9. On the external management device connected to the Routing Engine, monitor the startup process to verify that the system has booted properly.

NOTE: If the system is completely powered off when you power on the power supply, the Routing Engine boots as the power supply completes its startup sequence. Normally, the router boots from the Junos OS on the CompactFlash card.

After turning on a power supply, wait at least 60 seconds before turning it off.

Configuring Member IDs for the Virtual Chassis

To re-enable the Virtual Chassis, you must configure the member ID on both devices in the Virtual Chassis configuration.

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. Repeat Step 1 to 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
```

The router reboots in preparation for forming the Virtual Chassis. After the reboot, all MPCs remain powered off until the Virtual Chassis port connection is configured.

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.

Configuring Virtual Chassis Ports

Wait until the system is fully booted back up after setting the VC member IDs. VCP ports need to be defined in order for line cards to function properly in the VC configuration.

1. Log into the console port on member 0 and configure the first Virtual Chassis port that connects to member 1.

```
{master:member0-re0}
user@host> request virtual-chassis vc-port set fpc-slot number pic-slot number port number
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.
2. Log into the console port on member 1 and configure the first Virtual Chassis port that connects to member 0.

```
{master:member1-re0}
user@host> request virtual-chassis vc-port set fpc-slot number pic-slot number port number
vc-port successfully set
```

3. After establishing the VCP link, wait for all line cards and REs to come online. Confirm the availability of ports that will be used as VCP links by issuing the `show interface` command.

```
user@host> show interface xe-0/0/1
Physical interface: xe-0/0/1, Enabled, Physical link is Up
  Interface index: 49195, SNMP ifIndex: 591
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex, BPDU Error: None,
MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Disabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 00:1d:b5:f7:4e:e1, Hardware address: 00:1d:b5:f7:4e:e1
  Last flapped   : 2011-06-01 00:42:03 PDT (00:02:42 ago)
  Input rate     : 0 bps (0 pps)
```

```
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None

  Logical interface xe-0/0/1.0 (Index 73) (SNMP ifIndex 523)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Input packets : 0
    Output packets: 0
    Protocol eth-switch, MTU: 0
      Flags: Trunk-Mode
```

4. Configure the remaining VCP ports 1 at a time. Wait 30 seconds after setting each VCP port.

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

Completing the SCBE2 Upgrade

To complete the SCBE2 upgrade procedure:

1. Verify that the installation is successful and the SCBE2 is online by issuing the `show chassis environment cb` command:

```
user@host> show chassis environment cb 0
CB 0 status
State        Online
Temperature    30 degrees C / 86 degrees F
...
user@host> show chassis environment cb 1
CB 1 status
State        Online
Temperature    30 degrees C / 86 degrees F
...
```

   Other details, such as, temperature, power, etc are also displayed along with the state.

2. Verify that the fabric planes come online correctly by issuing the `show chassis fabric summary` command:

```
user@host> show chassis fabric summary
Plane   State    Uptime
 0      Online   2 days, 19 hours, 10 minutes, 9 seconds
 1      Online   2 days, 19 hours, 10 minutes, 9 seconds
...
```

3. Verify that the backup Routing Engine is back online by issuing the `show chassis routing-engine 1` command:

```
user@host> show chassis routing-engine 1
Routing Engine Status:
Slot 1:
Current State        Backup
...
```

4. Verify the SCBE2s before you finish by issuing the `show chassis hardware` command:

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
CB 0             REV 08   750-048307   CABC9829          Enhanced MX SCB 2
CB 1             REV 08   750-048307   CABC9828          Enhanced MX SCB 2
...
```

You see that the router now has SCBE2s.

RELATED DOCUMENTATION

Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms by Rebooting the Routing Engines | 209

Configuring Member IDs for a Virtual Chassis | 65

Example: Replacing a Routing Engine in a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms | 122

4

CHAPTER

Configuring Virtual Chassis Ports to Interconnect Member Devices

IN THIS CHAPTER

Guidelines for Configuring Virtual Chassis Ports

IN THIS SECTION

To interconnect the member routers in a *Virtual Chassis* for MX Series 5G Universal Routing Platforms, you must configure Virtual Chassis ports on 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: Starting with Junos OS Release 14.1, you can preconfigure ports that are currently unavailable for use. Although a Virtual Chassis port is unavailable for use as a standard network port, you can configure this port as a standard network port even after you configure it as a Virtual Chassis port. However, the router does not apply the configuration until you delete the Virtual Chassis port from the Virtual Chassis configuration.

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 links within all trunks.

  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.

> **NOTE**: Virtual Chassis ports should be spread over all power zones in each chassis to make the best use of physical redundancy in the router. Distributing VCP interfaces minimizes the risk of split-primary role if power zones are de-energized by PEM or power feed loss. With distributed VCP interfaces, survival of a single power zone prevents split-primary and other undesirable protocol primary role conditions from occurring until all zones are de-energized.

  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.

- When using a channelized configuration on MPC7E MRATE, MPC8E MRATE, or a MPC9E MRATE PIC QSFP interfaces for VCP connections between members, you must configure a VCP interface on channel 0 of each QSFP interface to activate the port.

Platform-Specific Virtual Chassis Port Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform:

Platform	Difference
MPC10E Line Cards installed in a MX Virtual Chassis	Virtual Chassis ports and access link ports are not supported. Only uplink (core facing interfaces) are supported.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting with Junos OS Release 14.1, you can preconfigure ports that are currently unavailable for use.

RELATED DOCUMENTATION

Virtual Chassis Components Overview | 7
Configuring Virtual Chassis Ports to Interconnect Member Routers or Switches | 168
Class of Service Overview for Virtual Chassis Ports | 181
Guidelines for Configuring Class of Service for Virtual Chassis Ports | 187
Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring Virtual Chassis Ports to Interconnect Member Routers or Switches

To interconnect the member routers in an MX Series Virtual Chassis, you must use the `request virtual-chassis vc-port set` command to configure network ports into Virtual Chassis ports on Modular Port

Concentrator/Modular Interface Card (MPC/MIC) interfaces. To interconnect the member switches into an EX9200 Virtual Chassis, you must use the `request virtual-chassis vc-port set` command to configure network ports into Virtual Chassis ports on line card interfaces. After the `request virtual-chassis vc-port set` is configured on both ends of the link, a Virtual Chassis port that is dedicated to the task of interconnecting member devices is created and the link can no longer be used 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 in an MX Series Virtual Chassis, the software displays a warning message that you are operating without a valid Virtual Chassis software license.

A software license is not needed to create an EX9200 Virtual Chassis.

To configure Virtual Chassis ports:

1. Configure the Virtual Chassis ports on the router or switch 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 devices 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 device 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 or switches are online, and the Virtual Chassis has formed, you can issue Virtual Chassis commands from the terminal window of the primary router or switch.

> **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 the 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 primary router or switch.

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

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

For an MX Series or 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

Guidelines for Configuring Virtual Chassis Ports | 166

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

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 or switch. 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.

A software license is not needed to create an EX9200 Virtual Chassis.

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

1. In the console window on the router or switch 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 or switch 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 or switches.

   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 or switch confirms that all Virtual Chassis ports have been deleted from both member devices.

   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 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 5G Universal Routing Platforms | 137
Configuring an EX9200 Virtual Chassis
Example: Deleting a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms | 138
Guidelines for Configuring Virtual Chassis Ports | 166

5

CHAPTER

Configuring Locality Bias to Conserve Bandwidth on Virtual Chassis Ports

IN THIS CHAPTER

Locality Bias in a Virtual Chassis

IN THIS SECTION

By default, member routers in an MX Series Virtual Chassis distribute egress traffic equally across all egress port links. Starting with Junos OS Release 14.1, if you want to conserve bandwidth across the internal Virtual Chassis ports, you can use locality bias to direct unicast transit traffic to egress links on the same (local) member router.

How Locality Bias Works

Locality bias conserves Virtual Chassis port bandwidth in a two-member MX Series Virtual Chassis by directing unicast transit traffic for equal-cost multipath (ECMP) groups and aggregated Ethernet bundles to egress links in the same (local) member router, provided that the local member router has an equal or larger number of available egress links than the remote member router. Because locality bias directs all of the traffic towards the local member router, Virtual Chassis ports do not use bandwidth.

However, if the number of available remote member router egress links exceeds the number of available local member router egress links, the system reduces the amount of traffic in the local member router by using a ratio that is based on the number of remote versus local links. The amount of traffic that the system does not direct toward egress links on the local member router is then split evenly across the egress links in the remote member router.

If either the local member router or remote member router do not have available egress links, then the traffic forwarding state across the Virtual Chassis ports does not change.

Locality Bias Percentages

The router uses the following algorithms to determine the percentage of traffic that is directed toward the local member router egress links, where *L* is the number of egress links on the local member router and *R* is the number of egress links on the remote member router.

BEST PRACTICE: To avoid possible traffic loss and oversubscription on egress interfaces, make sure you understand the utilization requirements, such as total and available bandwidth, for the local links in your network before changing the locality bias configuration.

- If *L >= R*, then *Locality Bias Percentage* = 100 percent and the local member router receives all egress traffic.

  For example, if the local member router and remote member router each contain one egress link, then the locality bias is 100 percent. The router directs all unicast transit traffic that is destined for an ECMP group or aggregated Ethernet bundle to the local member router.

- If *L < R*, then *Locality Bias Percentage* = 200 * (L / (R + L))

  For example, if the local member router (*L*) contains one link and the remote member router (*R*) contains two links, the locality bias percentage calculation is

  200 * (1 / (2 + 1)) = 66

  This means that the system directs 66 percent of the unicast transit traffic destined for an ECMP group or aggregated Ethernet bundle toward the local member router. The system splits the remaining 34 percent of the unicast transit traffic equally between the remote member router egress links. Each of the two remote egress links in the example receives 17 percent of the traffic.

  NOTE: The actual amount of traffic that the local member router receives can vary slightly from the percentages in the algorithm calculations.

- If *L = 0* or *R = 0*, then locality bias does not change the forwarding state.

For both the *L < R* and *L >= R* algorithms, locality bias percentages are recalculated on each line card whenever one of the aggregated Ethernet child links goes up or down, or whenever a link is added to or removed from an ECMP bundle.

NOTE: If an ECMP bundle has one or more child links that are aggregated Ethernet links, then those aggregated Ethernet child links are always considered remote unless *all* of the aggregated Ethernet child links are local.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting with Junos OS Release 14.1, if you want to conserve bandwidth across the internal Virtual Chassis ports, you can use locality bias to direct unicast transit traffic to egress links on the same (local) member router.

RELATED DOCUMENTATION

Guidelines for Configuring Locality Bias in a Virtual Chassis | 177
Configuring Locality Bias for a Virtual Chassis | 178

Guidelines for Configuring Locality Bias in a Virtual Chassis

Starting in Junos OS 14.1, you can configure locality bias in an MX Series Virtual Chassis. Consider the following guidelines when doing so:

- Make sure you know the total and available amounts of bandwidth in your network. Enabling locality bias when local egress links do not have enough bandwidth to support additional traffic can result in immediate traffic loss. The system does not prevent configurations that result in oversubscription.
- Locality bias directs only user traffic toward the local member router. Locality bias does not affect load balancing of control traffic within the Virtual Chassis.
- You cannot configure adaptive aggregated Ethernet load balancing and locality bias together.
- You cannot configure weighted load balancing and locality bias together.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS 14.1, you can configure locality bias in an MX Series Virtual Chassis.

RELATED DOCUMENTATION

Locality Bias in a Virtual Chassis | 175

Understanding Aggregated Ethernet Load Balancing

Configuring Locality Bias for a Virtual Chassis | 178

Configuring Locality Bias for a Virtual Chassis

Starting in Junos OS Release 14.1, configuring locality bias enables you to conserve Virtual Chassis port bandwidth, reduce infrastructure costs, and reduce network latency in a two-member MX Series Virtual Chassis. Locality bias works by directing unicast traffic for equal-cost multipath (ECMP) groups and aggregated Ethernet bundles to egress links in the same (local) member router in the Virtual Chassis rather than to egress links in the remote member router.

You can enable locality bias by including the `locality-bias` statement at the `[edit virtual-chassis]` hierarchy level.

BEST PRACTICE: To avoid possible traffic loss and oversubscription on egress interfaces, make sure you understand the utilization requirements for the local links in your network before changing the locality bias configuration.

To configure locality bias for an MX Series Virtual Chassis:

1. Specify that you want to enable locality bias in the Virtual Chassis.

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

2. Commit the configuration.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, configuring locality bias enables you to conserve Virtual Chassis port bandwidth, reduce infrastructure costs, and reduce network latency in a two-member MX Series Virtual Chassis.

RELATED DOCUMENTATION

Configuring Preprovisioned Member Information for a Virtual Chassis | 56
Locality Bias in a Virtual Chassis | 175
Guidelines for Configuring Locality Bias in a Virtual Chassis | 177

6

CHAPTER

Configuring Class of Service for Virtual Chassis Ports

IN THIS CHAPTER

Class of Service Overview for Virtual Chassis Ports

IN THIS SECTION

By default, all *Virtual Chassis* port interfaces in a Virtual Chassis for MX Series 5G Universal Routing Platforms 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

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 Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces in the following MX Series 5G Universal Routing Platforms with dual Routing Engines:

- MX240 Universal Routing Platform
- MX480 Universal Routing Platform
- MX960 Universal Routing Platform

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 5G Universal Routing Platforms" on page 188.

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

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 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 | **187**

Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 5G Universal Routing Platforms | **188**

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 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 | **181**

Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 5G Universal Routing Platforms | **188**

Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 5G Universal Routing Platforms

IN THIS SECTION

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

Before you begin:

- Configure a Virtual Chassis consisting of two MX Series routers.

  See "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69

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 17 on page 189 shows the nondefault CoS scheduler hierarchy configured in this example for the Virtual Chassis ports.

Table 17: 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 primary 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

IN THIS SECTION

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 90
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 10
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
```

Procedure

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 primary 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 90
user@host# set schedulers s-high-priority priority high excess-priority high transmit-rate
percent 10
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 90;
        priority medium-high;
        excess-priority high;
    }
    s-high-priority {
        transmit-rate percent 10;
        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 primary router, enter `commit` from configuration mode.

RELATED DOCUMENTATION

Class of Service Overview for Virtual Chassis Ports | 181

Guidelines for Configuring Class of Service for Virtual Chassis Ports | 187

7

CHAPTER

Configuring Redundancy Mechanisms on Aggregated Ethernet Interfaces in a Virtual Chassis

IN THIS CHAPTER

Redundancy Mechanisms on Aggregated Ethernet Interfaces in a Virtual Chassis

IN THIS SECTION

Starting in Junos OS Release 13.2, an MX Series Virtual Chassis configured with targeted traffic distribution for IP demux or VLAN demux subscribers on aggregated Ethernet interfaces supports three types of redundancy mechanisms: link redundancy, module redundancy, and chassis redundancy.

Link Redundancy in a Virtual Chassis

By default, the router uses *link redundancy*, also known as *port redundancy*, as the default redundancy mechanism for targeted distribution on aggregated Ethernet interfaces. With link redundancy, the router assigns backup links for a subscriber based on the link with the fewest number of subscribers.

In an MX Series *Virtual Chassis* configured with link redundancy, the primary link and backup link can be assigned on the same Modular Port Concentrator/Modular Interface Card (MPC/MIC) module, on different MPC/MIC modules in the same member router, or on different MPC/MIC modules in different member routers. This feature provides redundancy if a link in the MX Series Virtual Chassis configuration fails.

Because link redundancy is the default redundancy mechanism, no special configuration is required on the Virtual Chassis primary router to enable it.

Module Redundancy in a Virtual Chassis

You can configure *module redundancy*, also known as *Flexible PIC Concentrator (FPC) redundancy*, to provide redundancy if a module or a link fails. The router assigns backup links for the subscriber interface on a different MPC/MIC module from the primary link, based on the link with the fewest number of subscribers among the links on different modules.

In an MX Series Virtual Chassis configured with link redundancy, the router assigns the primary link and backup link to different MPC/MIC modules. For purposes of link selection, the router gives all MPC/MIC modules in the Virtual Chassis equal weight, and disregards the role (primary or backup) of the member router in which the MPC/MIC module is installed. The router uses an algorithm to assign the primary and backup links, and is as likely to assign a primary link to an MPC/MIC module in the Virtual Chassis primary router as it is to assign the primary link to an MPC/MIC module in the Virtual Chassis backup router.

Chassis Redundancy in a Virtual Chassis

Unlike link redundancy and module redundancy, which are supported on both standalone routers and Virtual Chassis member routers, chassis redundancy is available only for member routers in an MX Series Virtual Chassis configuration.

Chassis redundancy and module redundancy use the same algorithm for link assignment, with the exception that in a Virtual Chassis with chassis redundancy configured, the router assigns the backup link to an MPC/MIC module in a member router *other* than the router on which the primary link resides. For example, in a two-member MX Series Virtual Chassis, if the primary link for the aggregated Ethernet bundle is assigned to an MPC/MIC module in the Virtual Chassis primary router, the router assigns the backup link to an MPC/MIC module in the Virtual Chassis backup router.

Chassis redundancy provides protection if the MPC/MIC module containing the primary link fails. In this event, the subscriber connections fail over to the backup link on the MPC/MIC module in the other member router.

> **BEST PRACTICE**: We recommend that you do not configure both module (FPC) redundancy and chassis redundancy for the same aggregated Ethernet interface in an MX Series Virtual Chassis. If you do, module redundancy takes precedence over chassis redundancy.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
13.2	Starting in Junos OS Release 13.2, an MX Series Virtual Chassis configured with targeted traffic distribution for IP demux or VLAN demux subscribers on aggregated Ethernet interfaces supports three types of redundancy mechanisms: link redundancy, module redundancy, and chassis redundancy.

RELATED DOCUMENTATION

Targeted Traffic Distribution on Aggregated Ethernet Interfaces in a Virtual Chassis | 203

Configuring Module Redundancy for a Virtual Chassis | 198

Configuring Chassis Redundancy for a Virtual Chassis | 200

Configuring Module Redundancy for a Virtual Chassis

By default, a router or switch uses link redundancy for aggregated Ethernet interfaces (bundles) configured with targeted traffic distribution. Starting in Junos OS Release 13.2, as an alternative to using link redundancy, you can configure module redundancy, also known as FPC redundancy, for a Virtual Chassis configured with targeted traffic distribution for IP demux or VLAN demux subscribers on aggregated Ethernet interfaces.

In a Virtual Chassis, module redundancy assigns the primary link and backup link to *different* MPC/MIC modules or line cards, regardless of the Virtual Chassis role (primary or backup) of the member device in which the module is installed. Module redundancy provides redundancy protection if a module or a link in the Virtual Chassis fails.

Before you begin:

- Configure a Virtual Chassis consisting of two MX Series routers or two EX9200 switches.

  See "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69

- Ensure that the aggregated Ethernet bundle is configured *without* link protection.

  See *Configuring Aggregated Ethernet Link Protection*

To configure module redundancy:

1. Log in to the console on the primary device of the Virtual Chassis.
2. Specify that you want to configure the demux logical interface.

```
{master:member0-re0} [edit]
user@host# edit interfaces demux0 unit logical-unit-number
```

3. Enable targeted distribution for the interface.

```
{master:member0-re0} [edit intefaces demux0 unit logical-unit-number]
user@host# set targeted-distribution
```

4. Specify the aggregated Ethernet bundle for which you want to configure module redundancy.

```
{master:member0-re0} [edit]
user@host# edit interfaces aenumber aggregated-ether-options
```

5. Enable module (FPC) redundancy for the specified aggregated Ethernet bundle.

```
{master:member0-re0} [edit interfaces aenumber aggregated-ether-options]
user@host# set logical-interface-fpc-redundancy
```

BEST PRACTICE: We recommend that you do not configure both module (FPC) redundancy and chassis redundancy for the same aggregated Ethernet interface in the Virtual Chassis. If you do, module redundancy takes precedence over chassis redundancy.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
13.2	Starting in Junos OS Release 13.2, as an alternative to using link redundancy, you can configure module redundancy, also known as FPC redundancy, for a Virtual Chassis configured with targeted traffic distribution for IP demux or VLAN demux subscribers on aggregated Ethernet interfaces.

RELATED DOCUMENTATION

Configuring Chassis Redundancy for a Virtual Chassis

By default, the router uses link redundancy for aggregated Ethernet interfaces (bundles) configured with targeted traffic distribution. Starting in Junos OS Release 13.2, as an alternative to using link redundancy, you can configure chassis redundancy for an MX Series Virtual Chassis configured with targeted traffic distribution for IP demux or VLAN demux subscribers on aggregated Ethernet interfaces.

In an MX Series Virtual Chassis, chassis redundancy assigns the backup link to an MPC/MIC module in a member router *other* than the member router on which the primary link resides. For example, in a two-member MX Series Virtual Chassis where the primary link for the aggregated Ethernet bundle is on an MPC/MIC module in the primary router, chassis redundancy assigns the backup link for the bundle to an MPC/MIC module in the backup router. Chassis redundancy provides protection if the MPC/MIC module containing the primary link fails. In this event, the subscriber connections fail over to the backup link on the MPC/MIC module in the other member router.

Unlike link redundancy and module redundancy, each of which are supported for both standalone routers and Virtual Chassis member routers, chassis redundancy is available only for member routers in an MX Series Virtual Chassis.

Before you begin:

- Configure a Virtual Chassis consisting of two MX Series routers.

  See "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69

- Ensure that the aggregated Ethernet bundle is configured *without* link protection.

  See *Configuring Aggregated Ethernet Link Protection*

To configure chassis redundancy for an MX Series Virtual Chassis:

1. Log in to the console on the primary router of the Virtual Chassis.

2. Specify that you want to configure the demux logical interface.

```
{master:member0-re0} [edit]
user@host# edit interfaces demux0 unit logical-unit-number
```

3. Enable targeted distribution for the interface.

```
{master:member0-re0} [edit intefaces demux0 unit logical-unit-number]
user@host# set targeted-distribution
```

4. Specify the aggregated Ethernet bundle for which you want to configure chassis redundancy.

```
{master:member0-re0} [edit]
user@host# edit interfaces aenumber aggregated-ether-options
```

5. Enable module (FPC) redundancy for the specified aggregated Ethernet bundle.

```
{master:member0-re0} [edit interfaces aenumber aggregated-ether-options]
user@host# set logical-interface-chassis-redundancy
```

> **BEST PRACTICE**: We recommend that you do not configure both module (FPC) redundancy and chassis redundancy for the same aggregated Ethernet interface in an MX Series Virtual Chassis. If you do, module redundancy takes precedence over chassis redundancy.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
13.2	Starting in Junos OS Release 13.2, as an alternative to using link redundancy, you can configure chassis redundancy for an MX Series Virtual Chassis configured with targeted traffic distribution for IP demux or VLAN demux subscribers on aggregated Ethernet interfaces.

RELATED DOCUMENTATION

Targeted Traffic Distribution on Aggregated Ethernet Interfaces in a Virtual Chassis | 203

Multichassis Link Aggregation in a Virtual Chassis

Starting in Junos OS Release 13.3, you can configure multichassis link aggregation (MC-LAG) in an MX Series Virtual Chassis.

MC-LAG enables a device to form a logical link aggregation group interface with two or more other devices. The MC-LAG devices use the Inter-Chassis Communication Protocol (ICCP) to exchange control information between two MC-LAG network devices.

When you configure MC-LAG with an MX Series Virtual Chassis, the link aggregation group spans links to two Virtual Chassis configurations. Each Virtual Chassis consists of two MX Series member routers that form a logical system managed as a single network element. ICCP exchanges control information between the global primary router (VC-P) of the first Virtual Chassis and the VC-P of the second Virtual Chassis.

To configure MC-LAG on member routers in a Virtual Chassis, use the same procedure that you would use to configure MC-LAG on a standalone MX Series router.

NOTE: Internet Group Management Protocol (IGMP) snooping is not supported on MC-LAG interfaces in an MX Series Virtual Chassis.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
13.3	Starting in Junos OS Release 13.3, you can configure multichassis link aggregation (MC-LAG) in an MX Series Virtual Chassis.

RELATED DOCUMENTATION

Configuring Multichassis Link Aggregation on MX Series Routers

Global Roles and Local Roles in a Virtual Chassis | 14

Targeted Traffic Distribution on Aggregated Ethernet Interfaces in a Virtual Chassis

IN THIS SECTION

By default, member routers or switches in an EX9200 or MX Series *Virtual Chassis* use hash-based traffic distribution for subscriber interfaces in aggregated Ethernet bundles configured without link protection. The hash-based model distributes subscriber interface traffic over multiple links in the bundle, enabling you to load balance multiple traffic flows through the logical subscriber interface.

Starting in Junos OS Release 13.2, as an alternative to using hash-based distribution in an EX9200 or MX Series Virtual Chassis, you can configure targeted traffic distribution for IP demultiplexing (demux) or VLAN demux subscriber interfaces in an aggregated Ethernet bundle that is configured without link protection.

Targeted Distribution in a Virtual Chassis

Targeted distribution enables you to configure the Virtual Chassis to send (target) all egress data traffic for a logical subscriber interface across a single member link in an *aggregated Ethernet bundle*, also referred to as an IEEE 802.3ad link aggregation group (LAG) bundle. You configure targeted distribution for a demux subscriber interface on the Virtual Chassis primary router or switch.

With targeted distribution, the router or switch in a Virtual Chassis assigns the primary member link and backup member link for the aggregated Ethernet bundle across *all* Virtual Chassis port links that belong to the aggregated Ethernet bundle. To accomplish load balancing, the router or switch evenly distributes the demux subscriber interfaces over these member links.

Benefits of Targeted Distribution

Targeted distribution is especially useful in a Virtual Chassis configuration in which subscriber traffic enters through a Virtual Chassis port on one member router or switch and exits through a Virtual Chassis port on a different member router or switch. By combining Virtual Chassis ports from different member router or switches as member links of the aggregated Ethernet bundle, targeted distribution provides increased redundancy in the event of a chassis or link failure.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
13.2	Starting in Junos OS Release 13.2, as an alternative to using hash-based distribution in an EX9200 or MX Series Virtual Chassis, you can configure targeted traffic distribution for IP demultiplexing (demux) or VLAN demux subscriber interfaces in an aggregated Ethernet bundle that is configured without link protection.

RELATED DOCUMENTATION

Redundancy Mechanisms on Aggregated Ethernet Interfaces in a Virtual Chassis | 196
Configuring Module Redundancy for a Virtual Chassis | 198
Configuring Chassis Redundancy for a Virtual Chassis | 200

Understanding Support for Targeted Distribution of Logical Interface Sets of Static VLANs over Aggregated Ethernet Logical Interfaces

IN THIS SECTION

A logical interface set represents logical bundling of subscribers. Targeted distribution enables the selection of links for subscribers like VLAN demultiplexing (demux), PPPoE, or static VLANs that are managed using logical interfaces stacked over a logical interface set. When aggregated Ethernet is configured as the underlying interface for a logical interface set, the traffic from the logical interface set either flows according to a hash-based distribution or, if link-protection is configured on the aggregated Ethernet bundle, then it flows through one aggregated Ethernet link. Subscribers are managed by the underlying logical interfaces or interface sets and the traffic passes through the underlying aggregated Ethernet by choosing one of the member links from the bundle. Link selection is done per subscriber basis.

Manual targeted distribution for VLAN demux, PPPoE, and static VLANs over aggregated Ethernet logical interfaces requires explicit configuration of primary and backup member links for the VLANs. To overcome this overhead, Junos OS allows automatic targeted distribution for aggregated logical interfaces to automatically select the primary and the backup links and also to ensure that the load is balanced across all the member links. Figure 6 on page 205 shows configuration of static VLANs managed by logical interfaces ae0.0 and ae0.1 configured over logical interface set IFLset over aggregated Ethernet interfaces.

Figure 6: Targeted Distribution of Logical Interface sSet over Aggregated Ethernet Interfaces

By default, Junos OS allows automatic targeted distribution of static VLANs. For manual targeting, you configure targeting type as manual.

For a logical interface set whose underlying interface is an aggregated Ethernet interface, VLAN demux or PPPoE, the targeted distribution is configured and the DCD assigns one of the links of the aggregated Ethernet bundle as the primary link and another as the backup link for each of the logical interface set in order to evenly distribute the subscribers. The primary member link selection is based on the minimum primary load and the backup member link selection is based on the minimum overall load on the member links.

By default, the weight of the VLAN is 1 and the weight of the logical interface set is the accumulation of the underlying logical interfaces along with the default value of the logical interface set that is 1. Junos OS allows you to assign weight for all the targeted subscribers like VLAN demux, PPPoe, and static VLANs. The value of the weight assigned is based on the business, or class of service, or bandwidth requirement. The value of the weight ranges from 1 through 1000. The member links are then assigned based on the value of the weight. Targeted distribution is configured at the logical interface level or at logical interface set level or both. When logical interface set participates in targeting there is no special link selected for underlying logical interfaces. Link selected for logical interface set is propagated to underlying VLANs.

Consider upper as the logical interface set and lower as the underlying logical interfaces. When targeted distribution is configured, then the following applies:

- If the value of lower is set ,then value of upper has to be set. Upper can be set irrespective of whether lower is set or not.
- If the VLANs have configured weights and the logical interface set has no configured weight ,then the weight of the logical interface set is the accumulation of the VLAN weights.
- If the logical interface sets have configured weights, then the weights are used to distribute the logical interface sets irrespective of the weight of any member VLANs.

Rebalancing happens when the subscribers get added or removed from the bundle. The bundle is considered to be balanced as long as the difference of load on any two links in that bundle is not greater than the value of the highest weighted logical interface set or logical interface.

In CoS, there are two modes of operation with respect to aggregated Ethernet - scale and replicate. Scale mode allows distribution of load across the links while replicate mode load on all the links. The default scale mode is useful when all the links are active. When there are only two links active at a given point then replicate mode is more useful. Hence, Junos OS allows replicate mode for each member link on the bundle for static VLANs irrespective of whether it is auto or manual targeting.

Configuring Targeting Logical Interface Sets over Static VLAN Logical Interfaces

To configure targeted distribution of logical interface sets of static VLANs over aggregated Ethernet logical interfaces:

1. Configure the value of the weight of the member links for targeted distribution at the logical interface set level. When the weight is configured at the logical interface set level, then the weight is used to distribute logical interface sets and the weight of the member VLANs is ignored.

   ```
   [edit interfaces interface-set interface-set-name targeted-options]
   user@host# weight value;
   ```

   For example, configure the weight of the logical interface set *iflset* as 20.

   ```
   [edit interfaces interface-set iflset targeted-options]
   user@host# weight 20;
   ```

2. Configure the value of the weight for each logical interfaces for targeted distribution. The weight of the logical interface set is the accumulation of the weight of the logical interfaces.

```
[edit interfaces interface-name unit unit-number targeted-options]
user@host# weight value;
```

For example, configure the weight of logical interface ae0 as 20.

```
[edit interfaces interface ae0 targeted-options]
user@host# weight 20;
```

RELATED DOCUMENTATION

targeted-options

weight

8

CHAPTER

Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms by Rebooting the Routing Engines

IN THIS CHAPTER

Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms by Rebooting the Routing Engines

IN THIS SECTION

You can upgrade an MX Series Virtual Chassis configuration from Junos OS Release 11.2 to a later release by rebooting each of the Routing Engines. Both member routers in the Virtual Chassis must have dual Routing Engines installed.

> **NOTE**: Make sure all four Routing Engines in the Virtual Chassis (both Routing Engines in the primary 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 two-member MX Series Virtual Chassis by rebooting the Routing Engines. For information about upgrading Junos OS in an MX Series Virtual Chassis by performing a unified ISSU, see "Upgrading Junos OS in an MX Series Virtual Chassis by Performing a Unified ISSU" on page 224.

Requirements

This example uses the following software and hardware and components:

- Junos OS Release 12.3 and later releases
- One MX240 Universal Routing Platform with dual Routing Engines
- One MX480 Universal Routing Platform with dual Routing Engines

NOTE: This configuration example has been tested using the software release listed and is assumed to work on all later releases.

See Table 18 on page 212 for information about the hardware installed in each MX Series router.

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.

Overview and Topology

IN THIS SECTION

To upgrade Junos OS in an MX Series Virtual Chassis configuration by rebooting the Routing Engines, you must:

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

Topology

This example upgrades Junos OS in an MX Series Virtual Chassis configuration that uses the basic topology shown in Figure 7 on page 211. For redundancy, each member router is configured with two Virtual Chassis ports.

Figure 7: Sample Topology for a Virtual Chassis with Two MX Series Routers

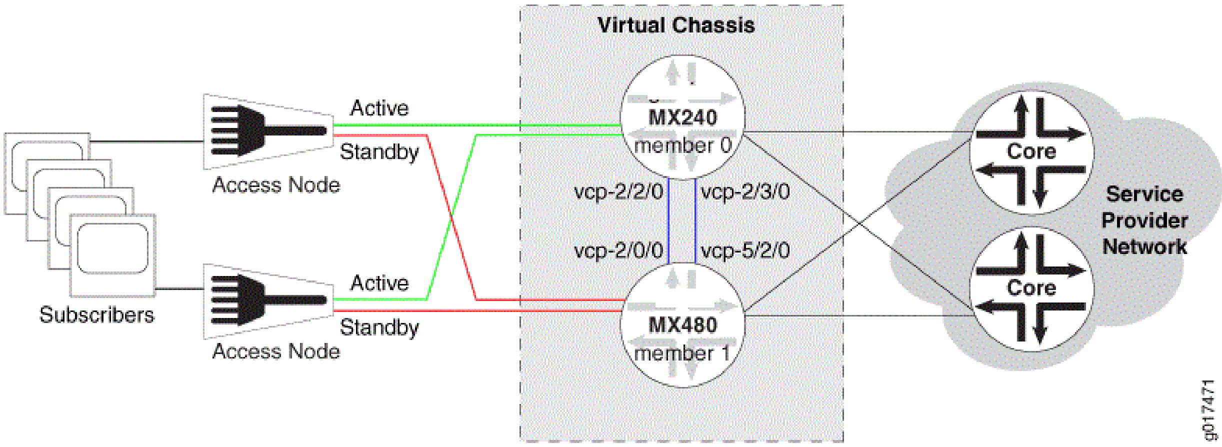


Table 18 on page 212 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 18: 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: • 60-Gigabit Ethernet Enhanced Queuing MPC • 20-port Gigabit Ethernet MIC with SFP • 4-port 10-Gigabit Ethernet MIC with XFP • Primary 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 (primary)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 18: 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: • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Primary 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

IN THIS SECTION

To upgrade Junos OS in a two-member MX Series Virtual Chassis by rebooting the Routing Engines, perform these tasks:

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 primary Routing Engine on the Virtual Chassis primary router (VC-P).

   See *Downloading Software*.

2. Disable nonstop active routing on the primary router.

```
{master:member0-re0}[edit routing-options]
user@gladius# deactivate nonstop-routing
```

3. Disable graceful Routing Engine switchover on the primary router.

```
{master:member0-re0}[edit chassis redundancy]
user@gladius# deactivate graceful-switchover
```

4. Commit the configuration on the primary router.

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 primary router (`member0-re0`) in the Virtual Chassis:

- Install the software package on VC-Pp.

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

  On a properly formed Virtual Chassis, this command propagates the image to all four Routing Engines.

Results

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

```
user@gladius> show version invoke-on all-routing-engines
member0-re0:
--------------------------------------------------------------------------
Hostname: gladius
Model: mx240
. . .
JUNOS Installation Software [14.1R1.10]

member0-re1:
--------------------------------------------------------------------------
Hostname:gladius1
Model: mx240
. . .
JUNOS Installation Software [14.1R1.10]

member1-re0:
--------------------------------------------------------------------------
```

```
Hostname: trefoil
Model: mx240
. . .
JUNOS Installation Software [14.1R1.10]

member1-re1:
--------------------------------------------------------------------------
Hostname: trefoil1
Model: mx240
. . .
JUNOS Installation Software [14.1R1.10]
```

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 primary router (`member0-re0`):

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

   ```
   {master:member0-re0}[edit chassis redundancy]
   user@gladius# activate graceful-switchover
   ```

2. Re-enable nonstop active routing on the primary router.

   ```
   {master:member0-re0}[edit routing-options]
   user@gladius# activate nonstop-routing
   ```

3. Commit the configuration on the primary router.

Rebooting the Routing Engines to Run the New Junos OS Release

Step-by-Step Procedure

NOTE: Rebooting both Routing Engines in the VC-P chassis may not result in a graceful switchover to the VC-B chassis, and is not recommended.

To reboot each of the four Routing Engines in an MX Series Virtual Chassis from the Virtual Chassis primary router (`member0-re0`):

- Use the `request system reboot` command with no options.

```
{master:member0-re0}
user@gladius> request system reboot
```

  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.

RELATED DOCUMENTATION

Upgrading Junos OS in an MX Series Virtual Chassis by Performing a Unified ISSU | 224
Interchassis Redundancy and Virtual Chassis Overview | 2
Virtual Chassis Components Overview | 7
Switchover Behavior in an MX Series Virtual Chassis | 27

9

CHAPTER

Upgrading Junos OS in an MX Series Virtual Chassis by Performing a Unified In-Service Software Upgrade (ISSU)

IN THIS CHAPTER

Unified ISSU in a Virtual Chassis

IN THIS SECTION

Starting in Junos OS Release 14.1, you can perform a unified in-service software upgrade (unified ISSU) for an MX Series Virtual Chassis configuration. Unified ISSU enables you to upgrade the Junos OS system software on the Virtual Chassis member routers with minimal traffic disruption and no disruption on the control plane.

NOTE: Starting in Junos OS Release 21.4R1 unified ISSU for MX-Virtual-Chassis is not supported and sequential ISSU model is supported. For details, see "How to Use Sequential Upgrade in an MX Series Virtual Chassis" on page 229

This topic assumes that you are familiar with the global roles and local roles in an MX Series Virtual Chassis. For information, see "Global Roles and Local Roles in a Virtual Chassis" on page 14.

Benefits of Performing a Unified ISSU in an Virtual Chassis

Performing a unified ISSU in an MX Series Virtual Chassis provides the following benefits:

- Upgrades the Junos OS software package while maintaining subscriber sessions.
- Reduces risk associated with a software upgrade. After performing a unified ISSU, the resulting system is exactly the same as if you had upgraded it with a system reboot.
- Prevents software upgrades from negatively affecting the service provider’s ability to fulfill strict service-level agreements (SLAs).
- Eliminates network downtime during software image upgrades.
- Enables faster implementation of new Junos OS features.

- Provides feature parity with unified ISSU support on standalone MX Series routers.

Prerequisites for Performing a Unified ISSU in a Virtual Chassis

Before you start a unified ISSU in a two-member MX Series Virtual Chassis, make sure you do all of the following:

- Ensure that all four Routing Engines in the Virtual Chassis (both Routing Engines in the primary router and both Routing Engines in the backup router) are running the same Junos OS software release.
- Back up the existing router configuration so you can revert (roll back) to it if necessary.
- Verify that both graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) are enabled.

How Unified ISSU Works in a Virtual Chassis

To perform a unified ISSU in an MX Series Virtual Chassis, you issue the `request system software in-service-upgrade` *package-name* command from the console window for the primary Routing Engine in the Virtual Chassis primary router (VC-Pp). Issuing this command from the VC-Pp copies the software package to all other Routing Engines in the Virtual Chassis.

The `request system software in-service-upgrade` *package-name* command functions the same for upgrading member routers in a Virtual Chassis configuration as it does for upgrading a standalone MX Series router with dual Routing Engines, *with the following exceptions*:

- The `no-copy`, `no-old-master-upgrade`, and `unlink` options for the `request system software in-service-upgrade` command are not available for an MX Series Virtual Chassis.
- The `reboot` option for the `request system software in-service-upgrade` command is accepted but ignored for an MX Series Virtual Chassis. A unified ISSU always reboots all Routing Engines in the Virtual Chassis member routers.

At a high level, the software performs the following actions after you issue the `request system software in-service-upgrade` *package-name* command to upgrade to a new Junos OS software release in a two-member Virtual Chassis configuration:

1. Arms the new Junos OS software release on all Routing Engines in the Virtual Chassis.

   The Routing Engines are still running the old Junos OS software release.

2. Upgrades both standby (backup) Routing Engines (VC-Ps and VC-Bs) in the Virtual Chassis.

   The Virtual Chassis is still actively forwarding traffic.

3. Performs a local switchover of the Routing Engines in the Virtual Chassis backup router (VC-B).

   The local switchover causes the VC-Bs upgraded in Step 2 to become the VC-Bp, and the VC-Bp that was still running the old Junos OS software to become the VC-Bs. The VC-Bp is now running the new Junos OS software release, and the VC-Bs is still running the old Junos OS software release. The Virtual Chassis is still actively forwarding traffic.

4. Upgrades the Packet Forwarding Engines to the new Junos OS software release.

   The Packet Forwarding Engines are now using the upgraded VC-Bp as the Virtual Chassis protocol primary.

5. Performs a local switchover of the Routing Engines in the Virtual Chassis primary router (VC-P).

   The local switchover of the VC-P also causes a global switchover in the Virtual Chassis, which causes the VC-P to become the VC-B. As a result, the VC-Pp becomes the VC-Bs, and the VC-Ps becomes the VC-Bp. The global switchover on the VC-B causes the VC-Bp to become the VC-Pp, and the VC-Bs to become the VC-Ps.

   The VC-Pp and VC-Bp are now running the new Junos OS software release. The VC-Ps (originally the VC-Bp) and VC-Bs (originally the VC-Pp) are still running the old Junos OS software release.

6. Upgrades the standby Routing Engines in the Virtual Chassis (VC-Ps and VC-Bs).

   The Virtual Chassis is now fully upgraded to the new Junos OS software release.

Virtual Chassis Role Transitions After a Unified ISSU

A unified ISSU in an MX Series Virtual Chassis upgrades all Routing Engines in the Virtual Chassis to the new Junos OS software release. In a two-member Virtual Chassis, this includes four Routing Engines: the primary and standby (backup) Routing Engines in the Virtual Chassis primary router, and the primary and standby Routing Engines in the Virtual Chassis backup router. As a result, the member routers and their associated Routing Engines undergo both global and local role transitions after the unified ISSU completes.

A *global role transition* changes the primary role in the Virtual Chassis by switching the global roles of the Virtual Chassis primary router (VC-P) and Virtual Chassis backup router (VC-B), and applies globally across the entire Virtual Chassis. A *local role transition* toggles the local primary and backup roles (`master` and `standby`, or `m` and `s`) of each of the two Routing Engines in a member router, and applies locally only to that member router.

A unified ISSU in an MX Series Virtual Chassis causes the global and local role transitions listed in Table 19 on page 222.

Table 19: Virtual Chassis Role Transitions After Unified ISSU

Virtual Chassis Role *Before* Unified ISSU	Virtual Chassis Role *After* Unified ISSU	Type of Role Change
Virtual Chassis primary router (VC-P)	Virtual Chassis backup router (VC-B)	Global
Virtual Chassis backup router (VC-B)	Virtual Chassis primary router (VC-P)	Global
Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Local
Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)	Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)	Local
Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)	Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)	Local
Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Primary Routing Engine in the Virtual Chassis primary router(VC-Pp)	Local

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can perform a unified in-service software upgrade (unified ISSU) for an MX Series Virtual Chassis configuration.

RELATED DOCUMENTATION

Preparing for a Unified ISSU in an MX Series Virtual Chassis | 223
Upgrading Junos OS in an MX Series Virtual Chassis by Performing a Unified ISSU | 224
Global Roles and Local Roles in a Virtual Chassis | 14

Unified ISSU System Requirements

Preparing for a Unified ISSU in an MX Series Virtual Chassis

Starting in Junos OS Release 14.1, you can perform a unified in-service software upgrade (unified ISSU) in an MX Series Virtual Chassis. Before you begin such an upgrade, perform the following tasks to help ensure its success.

> **NOTE**: For recommended settings on MX Virtual Chassis devices, please consult Maximizing Scaling and Performance for MX Series Virtual Chassis on the Juniper Networks Knowledge Base.

To prepare for a unified ISSU in a two-member MX Series Virtual Chassis:

1. Back up the existing system software to each member router's hard disk so you can roll back to it if necessary.

   Isssue the `request system snapshot all-members` command to archive data and executable areas for all members of the Virtual Chassis configuration.

2. Make sure enhanced IP network services is configured.

   See "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69.

3. Verify that both graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) are enabled.

   See "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69.

4. Make sure the router is configured to delay for 180 seconds (3 minutes) before removing access routes and access-internal routes for DHCP and PPP subscriber management after a graceful Routing Engine switchover takes place.

   See *Minimize Traffic Loss Due to Stale Route Removal After a Graceful Routing Engine Switchover*.

5. Ensure that the router is configured to use a *hold-time* value of 300 seconds when negotiating a BGP connection with the peer.

   See *BGP Messages Overview*.

6. Stop all CPU-intensive periodic operations running on the Virtual Chassis in order to conserve system resources.

Examples of such CPU-intensive operations include, but are not limited to, periodic SNMP Get and SNMP Walk requests, isssuing Junos OS operational mode commands with detailed or extensive output, and runnning Stylesheet Language Alternative Syntax (SLAX) operational scripts that monitor device status or network events.

7. (Aggregated Ethernet interfaces only) If you are using aggregated Ethernet interfaces, make sure each interface is configured to use a `slow` interval (every 30 seconds) for *periodic* transmission of Link Aggregation Control Protocol (LACP) packets.

   See Configuring LACP for Aggregated Ethernet Interfaces.

8. (IS-IS interfaces only) If you are using IS-IS interfaces configured with the `iso` protocol family, make sure the link-state PDU lifetime value (*lsp-lifetime*) is set to 65317 seconds.

   See *Example: Configuring the Transmission Frequency for Link-State PDUs on IS-IS Interfaces*.

BEST PRACTICE: For additional information about unified ISSU, consult our Knowledge Base.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can perform a unified in-service software upgrade (unified ISSU) in an MX Series Virtual Chassis.

RELATED DOCUMENTATION

Upgrading Junos OS in an MX Series Virtual Chassis by Performing a Unified ISSU | 224
Unified ISSU in a Virtual Chassis | 219

Upgrading Junos OS in an MX Series Virtual Chassis by Performing a Unified ISSU

Starting in Junos OS Release 14.1, you can upgrade the member routers in a two-member MX Series Virtual Chassis to a later release by performing a unified in-service software upgrade (unified ISSU). Unified ISSU enables you to upgrade to a new Junos OS software release with minimal traffic disruption and no loss of subscriber sessions.

The unified ISSU procedure upgrades and changes the roles of all Routing Engines in the Virtual Chassis to the new Junos OS software release. In a two-member Virtual Chassis, this includes four Routing Engines: the primary and standby Routing Engines in the Virtual Chassis primary router, and the primary and standby Routing Engines in the Virtual Chassis backup router.

This procedure describes how to upgrade Junos OS in an MX Series Virtual Chassis by performing a unified ISSU. For information about upgrading Junos OS in an MX Series Virtual Chassis by rebooting the Routing Engines, see "Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms by Rebooting the Routing Engines" on page 209.

Before you begin a unified ISSU in a two-member MX Series Virtual Chassis, perform the following tasks:

- Prepare the member routers for the unified ISSU operation.

  See "Preparing for a Unified ISSU in an MX Series Virtual Chassis" on page 223.

To perform a unified ISSU in a two-member MX Series Virtual Chassis:

1. Use FTP or a Web browser to download the Junos OS software package from the Juniper Networks Support Web site.

   See *Downloading Software*.
2. Open four console windows to access the console ports on each of the four Routing Engines in the Virtual Chassis:

   - Primary Routing Engine in the Virtual Chassis primary router (VC-Pp)
   - Standby Routing Engine in the Virtual Chassis primary router (VC-Ps)
   - Primary Routing Engine in the Virtual Chassis backup router (VC-Bp)
   - Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)

   Opening separate console windows enables you to monitor the progress of the unified ISSU on each of the Routing Engines.
3. (Optional but recommended) Confirm that the member routers are ready for the unified ISSU process.

   Issue the following commands in each console window:

   ```
   {master:member0-re0}
   user@host> show virtual-chassis vc-port | match “Configured”
   user@host> show system switchover local
   user@host> show system uptime local
   ```

   Issuing these commands provides the following information:

- The **show virtual-chassis vc-port | match "Configured"** command output confirms that all Virtual Chassis port interfaces are properly configured and operational.

- The **show system switchover local** command output confirms that the configuration database and kernel database are ready for a unified ISSU.

- The **show system uptime local** command output displays the date and time when this Routing Engine was last booted, and how long it has been running. It typically takes from 5 minutes to 15 minutes since the last switchover or system reboot for the Routing Engine to be ready for a unified ISSU.

4. Verify that all four Routing Engines in the Virtual Chassis are running the same Junos OS software release.

   Issue the `show version invoke-on all-routing-engines` command with the `all-members` option to display the hostnames and version information about the software running on all Routing Engines in the Virtual Chassis.

```
{master:member0-re0}
user@host> show version all-members invoke-on all-routing-engines
```

5. Initiate the unified ISSU process.

   In the console window on the VC-Pp, issue the `request system software in-service-upgrade` *`package-name`* command.

```
{master:member0-re0}
user@host> request system software in-service-upgrade package-name
```

> **NOTE**: You do not need to specify the `reboot` option for the `request system software in-service-upgrade` command because a unified ISSU in an MX Series Virtual Chassis always reboots all Routing Engines in the member routers.
>
> For MX series routers with RE-S-X6-64G Routing Engine, upgrade the guest Junos OS and the Linux VM host OS by issuing the `request vmhost software in-service-upgrade` *`package-name`* command.

6. Check the console windows for each Routing Engine to monitor the unified ISSU progress and determine when the upgrade is complete.

   The unified ISSU can take between 30 minutes and 90 minutes to complete, depending on the size of your configuration. When the unified ISSU completes, the login prompt appears in the console windows.

For an example of the output for a unified ISSU operation in an MX Series Virtual Chassis, see *request system software in-service-upgrade* and *request vmhost software in-service-upgrade*.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can upgrade the member routers in a two-member MX Series Virtual Chassis to a later release by performing a unified in-service software upgrade (unified ISSU).

RELATED DOCUMENTATION

Preparing for a Unified ISSU in an MX Series Virtual Chassis | 223

Unified ISSU in a Virtual Chassis | 219

Example: Performing a Unified ISSU

Unified ISSU System Requirements

Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 5G Universal Routing Platforms by Rebooting the Routing Engines | 209

10

CHAPTER

Upgrading Junos OS in an MX Series Virtual Chassis by Performing a Sequential Upgrade

IN THIS CHAPTER

How to Use Sequential Upgrade in an MX Series Virtual Chassis

IN THIS SECTION

Sequential Upgrade Overview

Starting in Junos OS Release 20.1R1, MX Series Virtual Chassis configurations can use sequential upgrade to install new software releases with minimal network downtime. The sequential upgrade process is an alternative to unified in-service software upgrade (ISSU) that installs a new release and reboots each Virtual Chassis member router one at a time. While the upgrade installs on one member router, the other member router continues to operate and handle network operations. This lets you upgrade to a new release with minimal disruption to the network.

NOTE: Sequential upgrades are not supported in MX Series Virtual Chassis with MPC10 series Modular Port Concentrators (MPCs).

Use this document to learn about sequential upgrade, how it works, and how to initiate an sequential upgrade on MX Series Virtual Chassis configurations.

Benefits of Performing a Sequential Upgrade in a MX Series Virtual Chassis

Performing a sequential upgrade in an MX Series Virtual Chassis provides the following benefits:

- Upgrades the Junos OS software package while maintaining subscriber sessions

- Minimizes network downtime during software image upgrades
- Avoids upgrading all Flexible PIC Concentrators (FPCs) and both chassis and at the same time

Sequential upgrade is an alternative to unified ISSU. Compared to ISSU, sequential upgrade offers the following benefits:

- Easier troubleshooting. Sequential upgrade applies the upgrade to the backup router first, giving you a window to check on the success of the upgrade and troubleshoot if necessary
- Ability to back out of an upgrade. With sequential upgrade, you can issue the `request virtual-chassis upgrade cancel` command after the backup router is upgraded, giving you the flexibility to back out of an upgrade and roll back to the original software version
- Lower resource requirements for sequential upgrade

Prerequisites for Performing a Sequential Upgrade in a MX Series Virtual Chassis

Before you start a sequential upgrade in a two-member MX Series Virtual Chassis, make sure you do all of the following:

- Ensure that all four Routing Engines in the Virtual Chassis (both Routing Engines in the primary router and both Routing Engines in the backup router) are running the same Junos OS software release.
- For minimum traffic disruption, make sure that both member routers are configured with symmetric network interface configurations so traffic can continue to run on all interfaces after switching from the primary router to the backup router.
- Ensure that your network is configured to enable moving all traffic from one member router to the other.
- Back up the existing router configuration.
- Verify that both graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) are enabled.
- Disable load throttling with the `set system services resource-monitor no-load-throttle` command. This will allow subscriber logins during the upgrade process.
- Download the software release package you want to upgrade to from Juniper's website at https://support.juniper.net/support/downloads/

Performing a Sequential Upgrade in a MX Series Virtual Chassis

To perform a sequential upgrade in an MX Series Virtual Chassis, follow these steps:

1. From the Virtual Chassis primary router, issue the following CLI command to initiate the upgrade process on the Virtual Chassis backup router:

   `request virtual-chassis upgrade protocol-backup` *`package-name`*

2. After the upgrade finishes on the backup router, issue the following CLI command from the backup router to begin the same upgrade process for the Virtual Chassis primary router:

   `request virtual-chassis upgrade protocol-master` *`package-name`*

> **NOTE**: If you want to cancel the sequential upgrade process, you can issue the `request virtual-chassis upgrade cancel` command on any Routing Engine after the Virtual Chassis backup router is updated and before the Virtual Chassis primary router is updated. After canceling the upgrade process, use the `request system software rollback` command to rollback to the previously installed package, if necessary.

How Sequential Upgrade Works in a MX Series Virtual Chassis

At a high level, the software performs the following actions after you issue the `request virtual-chassis upgrade protocol-backup` *`package-name`* command to upgrade to a new Junos OS software release in a two-member Virtual Chassis configuration:

1. Verifies that an upgrade is not already in progress and exits if an upgrade is already running.
2. Validates the software image using existing installation support.
3. Performs configuration validation.
4. Copies the software image to the backup router Routing Engines.
5. Activates the software image on the backup router Routing Engines.
6. Reboots the backup router Routing Engines.

   > **NOTE**: You can override the automatic rebooting of Routing Engines by adding the `halt-re` statement to the CLI command as follows: `request virtual-chassis upgrade protocol-backup halt-re` *`package-name`*. This causes the Routine Engines to halt and wait for the operator to reboot them. Use this command if you want to replace hardware components like FPCs while the router is shut down during the upgrade process.

7. Polls the backup router for FPC interface synchronization after the Routing Engines reboot.

After upgrading the Virtual Chassis backup router, the next step is to issue the `request virtual-chassis upgrade protocol-master` *`package-name`* command from the backup router CLI to initiate the upgrade process on Virtual Chassis primary router. The software performs the same actions as listed in steps 1-7 above, but for the primary router Routing Engines. After the primary router Routing Engines reboot, the software performs the following actions:

- Switches the role of the Virtual Chassis backup router to become the Virtual Chassis primary router.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.1	Starting in Junos OS Release 20.1R1, MX Series Virtual Chassis configurations can use sequential upgrade to install new software releases with minimal network downtime.

RELATED DOCUMENTATION

Interchassis Redundancy and Virtual Chassis Overview | 2
Global Roles and Local Roles in a Virtual Chassis | 14
request virtual-chassis upgrade

11
CHAPTER

Monitoring an MX Series Virtual Chassis

IN THIS CHAPTER

Accessing the Virtual Chassis Through the Management Interface

The management Ethernet interface (**fxp0**) on an MX Series router or EX9200 switch 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 device 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 a 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 primary (primary) Routing Engine in the primary router or switch (protocol primary) for the Virtual Chassis.
- To access a specific Routing Engine in an individual member router or switch of the Virtual Chassis, configure an IP address for one of the following 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 or switch may be useful.

To access a 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 primary Routing Engine in the Virtual Chassis primary router or switch.

```
{master:member0-re0}[edit]
user@host# set interfaces fxp0 unit 0 family inet address ip-address/prefix-length primary-
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 primary-only
```

- Configure a management IP address that accesses a specified Routing Engine in an individual member router or switch 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

Verifying the Status of Virtual Chassis Member Routers or Switches

IN THIS SECTION

Purpose

Verify that the member routers or switches in an MX Series or EX9200 Virtual Chassis are properly configured.

Action

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

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

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Verifying the Operation of Virtual Chassis Ports

IN THIS SECTION

Purpose

Verify that the Virtual Chassis ports in an MX Series or EX9200 Virtual Chassis are properly configured and operational.

Action

- To display the status of the Virtual Chassis ports for both member routers or switches 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 or switch 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 or switch on which you are issuing the command:

```
user@host> show virtual-chassis vc-port local
```

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Verifying Neighbor Reachability for Member Routers or Switches in a Virtual Chassis

IN THIS SECTION

Purpose

Verify that each member router or switch in an MX Series or EX9200 Virtual Chassis has a path to reach the neighbor devices to which it is connected.

Action

- To display neighbor reachability information for both member devices in the Virtual Chassis:

```
user@host> show virtual-chassis active-topology all-members
```

- To display neighbor reachability information for a specified member device in the Virtual Chassis:

```
user@host> show virtual-chassis active-topology member member-id
```

- To display neighbor reachability information for the member device on which you are issuing the command:

```
user@host> show virtual-chassis active-topology local
```

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis

IN THIS SECTION

Purpose

Verify that each hardware device in an MX Series Virtual Chassis or an EX9200 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 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Determining GRES Readiness in a Virtual Chassis Configuration

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

To determine whether the member routers or switches in a 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 primary router or switch (VC-Pp) 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 or switches are synchronized and ready for the GRES operation.

To determine whether the member routers or switches 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 primary router or switch (VC-Pp).

   ```
   {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 or switches 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, completes before the global GRES operation begins.

2. Review the results of the `request virtual-chassis routing-engine master switch check` command to determine whether the member routers or switches are ready for a GRES operation from a database synchronization perspective.

- If the member routers or switches are ready for GRES, the `request virtual-chassis routing-engine master switch check` command displays a message confirming GRES readiness. For example:

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

- If the member routers or switches 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 or switches.

RELATED DOCUMENTATION

Switchover Behavior in an MX Series Virtual Chassis | 27
Virtual Chassis Components Overview | 7
Global Roles and Local Roles in a Virtual Chassis | 14
Understanding Graceful Routing Engine Switchover

Viewing Information in the Virtual Chassis Control Protocol Adjacency Database

IN THIS SECTION

Purpose

View information about neighbors in the Virtual Chassis Control Protocol (VCCP) adjacency database for a Virtual Chassis configuration.

Action

- To display VCCP neighbor adjacency information for both member devices in the Virtual Chassis:

```
user@host> show virtual-chassis protocol adjacency all-members
```

- To display VCCP neighbor adjacency information for a specified member device 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 or switch:

```
user@host> show virtual-chassis protocol adjacency member member-id system-id
```

- To display VCCP neighbor adjacency information for the member device on which you are issuing the command:

```
user@host> show virtual-chassis protocol adjacency local
```

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Viewing Information in the Virtual Chassis Control Protocol Link-State Database

IN THIS SECTION

Purpose

View information about protocol data unit (PDU) packets in the Virtual Chassis Control Protocol (VCCP) link-state database for a Virtual Chassis configuration.

Action

- To display VCCP PDU information for both member routers or switches in the Virtual Chassis:

```
user@host> show virtual-chassis protocol database all-members
```

- To display VCCP PDU information for a specified member router or switch 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 or switch:

```
user@host> show virtual-chassis protocol database member member-id system-id
```

- To display VCCP PDU information for the member router or switch on which you are issuing the command:

```
user@host> show virtual-chassis protocol database local
```

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database

IN THIS SECTION

Purpose

View information in the Virtual Chassis Control Protocol (VCCP) database about Virtual Chassis port interfaces in the Virtual Chassis.

Action

- To display VCCP information about Virtual Chassis port interfaces for both member routers or switches:

```
user@host> show virtual-chassis protocol interface all-members
```

- To display VCCP information about Virtual Chassis port interfaces for a specified member router or switch:

```
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 or switch on which you are issuing the command:

```
user@host> show virtual-chassis protocol interface local
```

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Viewing Virtual Chassis Control Protocol Routing Tables

IN THIS SECTION

Purpose

View Virtual Chassis Control Protocol (VCCP) unicast and multicast routing tables for an MX Series Virtual Chassis configuration.

Action

- To display the VCCP unicast and multicast routing tables for both member routers in the Virtual Chassis:

```
user@host> show virtual-chassis protocol route all-members
```

- To display the VCCP unicast and multicast routing tables for a specified member router in the Virtual Chassis:

```
user@host> show virtual-chassis protocol route member member-id
```

- To display the VCCP unicast and multicast routing tables to the destination with the specified system ID:

```
user@host> show virtual-chassis protocol route destination-id
```

- To display the VCCP unicast and multicast routing tables to the destination with the specified system ID on the specified member router:

```
user@host> show virtual-chassis protocol route member member-id destination-id
```

- To display the VCCP unicast and multicast routing tables for the member router on which you are issuing the command:

```
user@host> show virtual-chassis protocol route local
```

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 50

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Viewing Virtual Chassis Control Protocol Statistics for Member Devices and Virtual Chassis Ports

IN THIS SECTION

Purpose

View Virtual Chassis Control Protocol (VCCP) statistics for one or both member routers or switches, or for a specified Virtual Chassis port interface, in a Virtual Chassis configuration.

Action

- To display VCCP statistics for both member routers or switches in the Virtual Chassis:

```
user@host> show virtual-chassis protocol statistics all-members
```

- To display VCCP statistics for a specified member router or switch in the Virtual Chassis:

```
user@host> show virtual-chassis protocol statistics member member-id
```

- To display VCCP statistics for a specified Virtual Chassis port interface:

```
user@host> show virtual-chassis protocol statistics vcp-slot/pic/port.logical-unit-number
```

- To display VCCP statistics for the member router or switch on which you are issuing the command:

```
user@host> show virtual-chassis protocol statistics local
```

RELATED DOCUMENTATION

Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | **50**

Configuring an EX9200 Virtual Chassis

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | **69**

Verifying and Managing the Virtual Chassis Heartbeat Connection

IN THIS SECTION

Purpose

Verify the configuration and operation of a heartbeat connection for an MX Series Virtual Chassis.

> **NOTE**: Starting in Junos OS Release 14.1, you can configure a heartbeat connection for an MX Series Virtual Chassis.

Action

- To clear heartbeat connection statistics counters and timestamp fields for all (both) member routers, the specified member router, or the member router on which you are issuing the command:

```
user@host> clear virtual-chassis heartbeat <all-members>
user@host> clear virtual-chassis heartbeat member member-id
user@host> clear virtual-chassis heartbeat local
```

- To view the heartbeat connection state for all (both) member routers, the specified member router, or the member router on which you are issuing the command:

```
user@host> show virtual-chassis heartbeat <all-members>
user@host> show virtual-chassis heartbeat member member-id
user@host> show virtual-chassis heartbeat local
```

- To display and review the statistics collected by the heartbeat connection for all (both) member routers, the specified member router, or the member router on which you are issuing the command:

```
user@host> show virtual-chassis heartbeat detail <all-members>
user@host> show virtual-chassis heartbeat detail member member-id
user@host> show virtual-chassis heartbeat detail local
```

- To verify use (status) of the heartbeat connection when an adjacency disruption or split is detected in the Virtual Chassis:

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

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can configure a heartbeat connection for an MX Series Virtual Chassis.

RELATED DOCUMENTATION

Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in the Same Subnet | 259
Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in Different Subnets | 274
Configuring a Virtual Chassis Heartbeat Connection | 17

Inline Flow Monitoring for Virtual Chassis Overview

IN THIS SECTION

Inline flow monitoring enables you to monitor the flow of traffic by means of a router or switch participating in a network.

Inline flow monitoring for an MX Series Virtual Chassis or an EX9200 Virtual Chassis supports the following features:

- Active sampling and exporting of both IPv4 and IPv6 traffic flows. Active (inline) sampling occurs on an inline data path without the need for a services DPC.
- Sampling traffic flows in both the ingress and egress directions.
- Configuring flow collection on either IPv4 or IPv6 devices.
- Using the IPFIX flow collection template for traffic sampling. The IPFIX template supports both IPv4 and IPv6 export records.
- Sampling and exporting of VPLS flows
- Exporting of data in Version-IPFIX and Version 9 formats

Consider the following guidelines when you configure Virtual Chassis for inline flow monitoring.

Syntax of the sampling-instance Statement

To associate a sampling instance with an FPC in the Virtual Chassis primary router (member ID 0), use the `sampling-instance` *`instance-name`* statement at the `[edit chassis member` *`member-number`* `fpc slot` *`slot-number`*`]` hierarchy level, where *`member-number`* is 0 (zero) and *`slot-number`* is a number in the range 0 through 11. For example, the following statement associates a sampling instance named sample1 to the FPC in slot 1 of a Virtual Chassis primary router:

```
[edit chassis member 0 fpc slot 1]user@host# set sampling-instance sample1
```

To associate a sampling instance with an FPC in the Virtual Chassis backup router (member ID 1), use the `sampling-instance` *`instance-name`* statement at the `[edit chassis member` *`member-number`* `fpc slot` *`slot-number`*`]` hierarchy level, where *`member-number`* is 1 and *`slot-number`* is a number in the range 0 through 11. For example, the following statement associates a sampling instance named sample2 to the FPC in slot 2 of Virtual Chassis backup router:

```
[edit chassis member 1 fpc slot 2]user@host# set sampling-instance sample2
```

FPC Slot Numbers for the Virtual Chassis

After you configure the member ID and, optionally, slot count for each router that you want to add to the Virtual Chassis, the Routing Engines in that chassis reboot and the slots for line cards (FPCs) are renumbered. The FPC slot numbering used for each member router is based on the slot count and offsets used in the Virtual Chassis instead of the physical slot numbers where the line card is actually installed.

For example, assume that in your Virtual Chassis configuration, member 0 is an MX960 router and member 1 is an MX2010 router, with the default slot count (12) in effect on both routers. In this topology, a 10-Gigabit Ethernet interface that appears as xe-14/2/2 (FPC slot 14, PIC slot 2, port 2) in the `show services accounting status inline-jflow` command output or 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 offset of 12 for member 1.

NOTE: Platform support and associated slots depend on the Junos OS release in your installation.

For more information about how slot count and slot numbering work in an MX Series Virtual Chassis, see "Virtual Chassis Components Overview" on page 7.

RELATED DOCUMENTATION

Configuring Inline Active Flow Monitoring Using Routers, Switches or NFX250

Configuring Sampling Instance on MX, M and T Series Routers or QFX Series Switches

no-split-detection

Disabling Split Detection in a Virtual Chassis Configuration

Split Detection Behavior in a Virtual Chassis

Managing Files on Virtual Chassis Member Routers or Switches

In a Virtual Chassis configuration for MX Series 5G Universal Routing Platforms or EX9200 switches, you can manage files on local and remote member routers or switches 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 or switch 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 or switch and a specific Routing Engine:

- 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 or switch 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 device.

```
{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 | 2
Virtual Chassis Components Overview | 7
Format for Specifying Filenames and URLs in Junos OS CLI Commands

Virtual Chassis SNMP Traps

Junos OS supports the use of SNMP traps to monitor the routers, switches, and other devices in your network.

MX Virtual Chassis supports the following enterprise-specific traps:

- `jnxVccpPortUp`
- `jnxVccpPortDown`

An unexpected SNMP trap of `jnxVccpPortDown` with a `jnxVirtualChassisPortOperStatus` value of 2 (down) should be treated as a critical notification, as it indicates one or more of the VCP-interfaces has undergone a failure, and may result in degraded performance or decreased resiliency to further errors until the status is cleared by a `jnxVccpPortUp` with `jnxVirtualChassisPortOperStatus` value equal to 1 (up).

A detailed description may be found in the MIB, located https://www.juniper.net/documentation/en_US/junos15.1/topics/reference/mibs/mib-jnx-virtualchassis.txt.

RELATED DOCUMENTATION

Virtual Chassis Slot Number Mapping for Use with SNMP | 256
Virtual Chassis Components Overview | 7
SNMP MIB Explorer

Virtual Chassis Slot Number Mapping for Use with SNMP

Junos OS supports the use of SNMP to monitor the routers, switches, 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 or a member switch of an EX9200 Virtual Chassis, keep in mind that the offset used for slot numbering in the Virtual Chassis affects the value that appears for the jnxFruSlot object.

Table 20 on page 257 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 EX9200 or MX Series Virtual Chassis. For example, a jnxFruSlot value of 15 corresponds to physical slot 3 in member 0 of the Virtual Chassis. A jnxFruSlot value of 30 corresponds to physical slot 6 in member 1 of the Virtual Chassis.

Table 20: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series or EX9200 Virtual Chassis

jnxFruSlot Number	Line Card Slot Number	MX Series or EX9200 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
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

Table 20: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series or EX9200 Virtual Chassis *(Continued)*

jnxFruSlot Number	Line Card Slot Number	MX Series or EX9200 Virtual Chassis Member ID
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 | 7

SNMP MIB Explorer

Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in the Same Subnet

IN THIS SECTION

A *heartbeat connection* is an IP-based, bidirectional packet connection in an MX Series Virtual Chassis between the Virtual Chassis primary and backup routers. The *heartbeat packets* exchanged over this connection provide critical information about the availability and health of each member router. Starting in Junos OS Release 14.1, you can configure a heartbeat connection in an MX Series Virtual Chassis.

This example describes how to configure a heartbeat connection in an MX Series Virtual Chassis when both member routers reside in the same subnet. For information about configuring a heartbeat connection when the Virtual Chassis member routers reside in different subnets, see "Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in Different Subnets" on page 274.

Requirements

This example uses the following software and hardware components:

- Junos OS Release 14.1 and later releases
- Two MX240 Universal Routing Platforms, each with dual Routing Engines

This configuration example has been tested using the software release listed and is assumed to work on all later releases.

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.

Before you configure a heartbeat connection for a Virtual Chassis:

- Configure a Virtual Chassis consisting of two MX Series routers.

  See "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69

  As part of the preprovisioned Virtual Chassis configuration shown in the configuration example, you must create and apply the `member0-re0`, `member0-re1`, `member1-re0`, and `member1-re1` configuration groups for each member Routing Engine. Each configuration group includes a unique IP address for the management Ethernet interface (`fxp0`) on each Routing Engine.

  NOTE: When you create the preprovisioned Virtual Chassis configuration at the `[edit virtual-chassis]` hierarchy level, make sure you do *not* configure the `no-split-detection` statement to disable detection of a split in the Virtual Chassis. Using the `no-split-detection` statement is prohibited when you configure a Virtual Chassis heartbeat connection, and doing so causes the commit operation to fail.

- Ensure TCP connectivity between the primary Routing Engine in the Virtual Chassis primary router (VC-Pp) and the primary Routing Engine in the Virtual Chassis backup router (VC-Bp).

  The Virtual Chassis heartbeat connection opens a proprietary TCP port numbered 33087 on the VC-Pp to listen for heartbeat messages. If your network design includes firewalls or filters, make sure the network allows traffic between TCP port 33087 on the VC-Pp and the dynamically allocated TCP port on the VC-Bp.

Overview

IN THIS SECTION

A *heartbeat connection* is an IP-based, bidirectional packet connection between the primary router and backup router in an MX Series Virtual Chassis. The member routers forming the heartbeat connection exchange *heartbeat packets* that provide critical information about the availability and health of each member router. During a disruption or split in the Virtual Chassis configuration, the heartbeat connection prevents the member routers from changing primary role roles unnecessarily, which can cause undesirable results.

This example configures a heartbeat connection for an MX Series Virtual Chassis in which both MX240 member routers reside in the 10.4.0.0 subnet. Member router `master-router` is the global primary router for the Virtual Chassis (VC-P), and member router `backup-router` is the global backup router (VC-B). Both member routers have dual Routing Engines installed, and the heartbeat connection is configured between the primary Routing Engine in `master-router` (represented by VC-Pp or `member0-re0`) and the primary Routing Engine in `backup-router` (represented by VC-Bp or `member1-re0`).

Configuring a heartbeat connection for an MX Series Virtual Chassis when both member routers reside in the same subnet consists of the following tasks:

1. Configure the global `master-only` IP address for the `fxp0` management interface on the same subnet as the four Routing Engines in the Virtual Chassis.

2. Configure a network path for the heartbeat connection.

   This example uses a global static route to provide a path for member routers in the same subnet to reach each other via a TCP/IP connection.

3. Configure the global `master-only` IP address for the `fxp0` management interface as the heartbeat address to establish the Virtual Chassis heartbeat connection.

4. (Optional) Configure a nondefault value for the Virtual Chassis heartbeat timeout interval.

Topology

This example configures a heartbeat connection for an MX Series Virtual Chassis with both member routers in the same subnet. For redundancy, each member router is configured with two Virtual Chassis ports.

Table 21 on page 262 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 21: Components of the Sample MX Series Virtual Chassis with Member Routers in Same Subnet

Router Name	Hardware	Serial Number	Member ID	Role	Virtual Chassis Ports	Subnet
master-router	MX240 router with: • Primary 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`)	JN11026AFAFC	0	routing-engine (primary)	vcp-1/0/0 vcp-1/1/0	10.4.0.0/16
backup-router	MX240 router with: • Primary 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`)	JN112C2FCAFC	1	routing-engine (backup)	vcp-2/0/0 vcp-2/1/0	10.4.0.0/16

Configuration

IN THIS SECTION

To configure a heartbeat connection in an MX Series Virtual Chassis with both member routers in the same subnet, perform these tasks:

CLI Quick Configuration

To quickly configure a heartbeat connection for a Virtual Chassis with both member routers in the same subnet, copy the following commands and paste them into the router terminal window:

```
[edit]
set groups member0-re0 interfaces fxp0 unit 0 family inet address 10.4.2.210/16 master-only
set groups member0-re1 interfaces fxp0 unit 0 family inet address 10.4.2.210/16 master-only
set groups member1-re0 interfaces fxp0 unit 0 family inet address 10.4.2.210/16 master-only
set groups member1-re1 interfaces fxp0 unit 0 family inet address 10.4.2.210/16 master-only
set groups global routing-options static route 10.4.0.0/16 next-hop 10.4.0.1
set groups global routing-options static route 10.4.0.0/16 retain
set groups global routing-options static route 10.4.0.0/16 no-readvertise
set virtual-chassis heartbeat-address 10.4.2.210
set virtual-chassis heartbeat-timeout 20
```

Configuring a Consistent Management IP Address for Each Routing Engine

Step-by-Step Procedure

In addition to configuring a unique IP address for the `fxp0` management interface on each Routing Engine when you first set up the Virtual Chassis, you must configure an additional management IP address, known as the `master-only` address, to ensure consistent access to the `fxp0` management interface on the primary Routing Engine in the Virtual Chassis primary router (VC-Pp, represented by `member0-re0` in this example). You then use the `master-only` address as the heartbeat address to establish the Virtual Chassis heartbeat connection.

Because the Virtual Chassis member routers in this example both reside in the same subnet (10.4.0.0), you can configure the same `master-only` address for each Routing Engine. The `master-only` address is active only on the management interface for the VC-Pp. During a switchover, the `master-only` address moves to the new Routing Engine currently functioning as the VC-Pp.

To configure the `master-only` `fxp0` IP address for each Routing Engine:

- From the console on member 0, configure the same IP address for the `fxp0` management interface on each Routing Engine.

```
{master:member0-re0}[edit]
user@master-router# set groups member0-re0 interfaces fxp0 unit 0 family inet address
10.4.2.210/16 primary-only
user@master-router# set groups member0-re1 interfaces fxp0 unit 0 family inet address
10.4.2.210/16 master-only
user@master-router# set groups member1-re0 interfaces fxp0 unit 0 family inet address
10.4.2.210/16 master-only
user@master-router# set groups member1-re1 interfaces fxp0 unit 0 family inet address
10.4.2.210/16 master-only
```

Results

From the console on the Virtual Chassis primary router, display the results of the configuration for each configuration group. For brevity, portions of the configuration unrelated to this procedure are replaced by an ellipsis (...).

For `member0-re0`:

```
{master:member0-re0}[edit]
user@master-router# show groups member0-re0
system {
    host-name master-router;
    backup-router 10.4.0.1 destination [ 172.16.0.0/12 ... 10.204.0.0/16 ];
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
                address 10.4.2.100/16;
```

```
                address 10.4.2.210/16 {
                    master-only;
                }
            }
        }
    }
}
```

For `member0-re1`:

```
{master:member0-re0}[edit]
user@master-router# show groups member0-re1
system {
    host-name master-router1;
    backup-router 10.4.0.1 destination [ 172.16.0.0/12 ... 10.204.0.0/16 ];
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
                address 10.4.2.101/16;
                address 10.4.2.210/16 {
                    master-only;
                }
            }
        }
    }
}
```

For `member1-re0`:

```
{master:member0-re0}[edit]
user@master-router# show groups member1-re0
system {
    host-name backup-router;
    backup-router 10.4.0.1 destination [ 172.16.0.0/12 ... 10.204.0.0/16 ];
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
```

```
                address 10.4.3.101/16;
                address 10.4.2.210/16 {
                    master-only;
                }
            }
        }
    }
}
```

For `member1-re1`:

```
{master:member0-re0}[edit]
user@master-router# show groups member1-re1
system {
    host-name backup-router1;
    backup-router 10.4.0.1 destination [ 172.16.0.0/12 ... 10.204.0.0/16 ];
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
                address 10.4.3.102/16;
                address 10.4.2.210/16 {
                    master-only;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring a Static Route Between the Primary and Backup Member Routers

Step-by-Step Procedure

You must configure a secure and reliable path between the primary router and backup router for the exchange of TCP/IP heartbeat packets. The heartbeat packets provide critical information about the availability and health of each member router.

The route you create for the heartbeat connection must be independent of the Virtual Chassis port links. Specifically, you must ensure that the primary Routing Engine in the Virtual Chassis backup router

(VC-Bp) can make a TCP/IP connection to the `master-only` IP address of the primary Routing Engine in the Virtual Chassis primary router (VC-Pp).

This examples creates a global static route between the member routers to configure the heartbeat path. However, you can choose the method that best meets your needs to configure the heartbeat path for member routers in the same subnet. For example, you might use the member router's default gateway for this purpose.

> **BEST PRACTICE**: We recommend that you use the router management interface (`fxp0`) as the heartbeat path. The management interface is generally available earlier than the line card interfaces, and is typically connected to a more secure network than the other interfaces.

To create a static route between the primary and backup member routers:

- From the console on member 0, configure a static route between the member routers in subnet 10.4.0.0.

```
{master:member0-re0}[edit]
user@master-router# set groups global routing-options static route 10.4.0.0/16 next-hop 
10.4.0.1
user@master-router# set groups global routing-options static route 10.4.0.0/16 retain
user@master-router# set groups global routing-options static route 10.4.0.0/16 no-readvertise
```

Results

Display the results of the configuration. For brevity, portions of the configuration unrelated to this procedure are replaced by an ellipsis (...).

```
{master:member0-re0}[edit]
user@master-router# show groups global routing-options static
route 10.4.0.0/16 {
    next-hop 10.4.0.1;
    retain;
    no-readvertise;
}
...
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Heartbeat Address and Heartbeat Timeout

Step-by-Step Procedure

To establish the heartbeat connection in a two-member MX Series Virtual Chassis, you must configure the IP address for the connection between the primary and backup member routers. To ensure consistent access to the primary Routing Engine in the Virtual Chassis primary router (VC-Pp) regardless of which Routing Engine is currently active, you set the heartbeat address to the previously configured global `master-only` IP address for the `fxp0` management interface.

Optionally, you can also configure a nondefault value for the heartbeat timeout interval. The heartbeat timeout is the maximum time within which a Virtual Chassis member router must respond to a heartbeat packet sent by the other member router. If you do not explicitly configure the heartbeat timeout interval, the default value (2 seconds) applies.

To configure the heartbeat address and heartbeat timeout:

1. From the console on member 0, specify that you want to edit the Virtual Chassis preprovisioned configuration.

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

2. Configure the common `master-only` IP address for the `fxp0` management interface as the heartbeat address.

```
{master:member0-re0}[edit virtual-chassis]
user@master-router# set heartbeat-address 10.4.2.210
```

3. (Optional) Configure a nondefault value for the heartbeat timeout interval.

```
{master:member0-re0}[edit virtual-chassis]
user@master-router# set heartbeat-timeout 20
```

Results

Display the results of the configuration.

```
{master:member0-re0}[edit]
user@master-router# show virtual-chassis
preprovisioned;
traceoptions {
    file VCCP size 100m;
    flag all;
}
heartbeat-address 10.4.2.210;
heartbeat-timeout 20;
member 0 {
    role routing-engine;
    serial-number JN11026AFAFC;
}
member 1 {
    role routing-engine;
    serial-number JN112C2FCAFC;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

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

Verifying the Virtual Chassis Heartbeat Connection

Purpose

Verify that the heartbeat connection between the Virtual Chassis member routers is properly configured and operational.

Action

Display the state of one or both member routers when a heartbeat connection is configured.

```
{master:member0-re0}
user@master-router> show virtual-chassis heartbeat
member0:
---------------------------------------------------------------------------
Local        Remote          State        Time
10.4.2.210   10.4.3.101      Alive        2014-02-18 11:18:14 PST

member1:
---------------------------------------------------------------------------
Local        Remote          State        Time
10.4.3.101   10.4.2.210      Alive        2014-02-18 11:18:15 PST
```

Meaning

For each member router, the command output displays the IP addresses of the local and remote member routers that form the heartbeat connection. The value `Alive` in the `State` field confirms that the primary Routing Engine in the specified member router is connected and has received a heartbeat response message. The `Time` field specifies the date and time of the last connection state change.

Verifying Use of the Heartbeat Connection During an Adjacency Split or Disruption

Purpose

Verify use of the heartbeat connection when an adjacency disruption or split is detected in the Virtual Chassis.

Action

Display the status of the member routers in the Virtual Chassis:

```
{master:member0-re0}
user@master-router> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: 4806.94d6.2362

                                              Mastership          Neighbor List
Member ID       Status   Serial No    Model    priority    Role      ID  Interface

0 (FPC   0- 11) Heartbt  JN11026AFAFC mx240         129   Master*    1  vcp-1/0/0
                                                                     1  vcp-1/1/0

1 (FPC  12- 23) Prsnt    JN112C2FCAFC mx240         129   Backup     0  vcp-2/0/0
                                                                     0  vcp-2/1/0
```

Meaning

The `Status` field for member ID 0 displays `Heartbt`, which indicates that this member router has used the heartbeat packet connection to maintain primary role roles during an adjacency disruption or split in the Virtual Chassis configuration. The `Status` field for member ID 1 displays `Prsnt`, which indicates that this member router is connected to the Virtual Chassis.

If a router is not currently connected to the Virtual Chassis, the `Status` field displays `NotPrsnt`.

Verifying Virtual Chassis Member Health from Heartbeat Statistics

Purpose

Use statistics collected by the heartbeat connection to verify the availability and health of each Virtual Chassis member router. You can also use the `show virtual-chassis heartbeat detail` command to determine the maximum latency and minimum latency in your network.

Action

Display and review the statistics collected by the heartbeat connection.

```
{master:member0-re0}
user@master-router> show virtual-chassis heartbeat detail
member0:
--------------------------------------------------------------------------
Local           Remote          State       Time
10.4.2.210      10.4.3.101      Alive       2014-02-18 11:18:14 PST

Heartbeat statistics
  Heartbeats sent: 10079
  Heartbeats received: 10079
  Heartbeats lost/missed: 0
  Last time sent: 2014-02-18 20:03:10 PST (00:00:00 ago)
  Last time received: 2014-02-18 20:03:10 PST (00:00:00 ago)
  Maximum latency (secs): 0
  Minimum latency (secs): 0

member1:
--------------------------------------------------------------------------
Local           Remote          State       Time
10.4.3.101      10.4.2.210      Alive       2014-02-18 11:18:15 PST

Heartbeat statistics
  Heartbeats sent: 10083
  Heartbeats received: 10083
  Heartbeats lost/missed: 0
  Last time sent: 2014-02-18 20:03:09 PST (00:00:01 ago)
  Last time received: 2014-02-18 20:03:09 PST (00:00:01 ago)
  Maximum latency (secs): 0
  Minimum latency (secs): 0
```

Meaning

In this example, the number of heartbeat request messages sent (`Heartbeats sent`) equals the number of heartbeat response messages received (`Heartbeats received`), with no heartbeat messages lost (`Heartbeats lost/missed`). This indicates that both member routers forming the heartbeat connection were available and operational. Any difference between `Heartbeats sent` and `Heartbeats received` appears in the `Heartbeats lost/missed` field.

The `Maximum latency` and `Minimum latency` fields measure the maximum and minimum number of seconds that elapse on the local router between trasnmission of a heartbeat request message and receipt of a heartbeat response message. In this example, the value `0` in the `Maximum latency` and `Minimum latency` fields indicates that there is no measurable network delay caused by this operation. You can use the `Maximum latency` value to determine whether you need to increase the `heartbeat-timeout` to a value higher than the default (2 seconds). If the maximum latency in your network is too high to accommodate a 2-second `heartbeat-timeout` value, increasing the `heartbeat-timeout` interval enables you to account for network delay when a Virtual Chassis adjacency disruption or split occurs.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can configure a heartbeat connection in an MX Series Virtual Chassis.

RELATED DOCUMENTATION

Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in Different Subnets | 274

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring a Virtual Chassis Heartbeat Connection | 17

Global Roles and Local Roles in a Virtual Chassis | 14

Configuring a Consistent Management IP Address

Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in Different Subnets

IN THIS SECTION

A *heartbeat connection* is an IP-based, bidirectional packet connection in an MX Series Virtual Chassis between the Virtual Chassis primary and backup routers. The *heartbeat packets* exchanged over this connection provide critical information about the availability and health of each member router. Starting in Junos OS Release 14.1, you can configure a heartbeat connection in an MX Series Virtual Chassis.

This example describes how to configure a heartbeat connection in an MX Series Virtual Chassis when the member routers reside in different subnets. For information about configuring a heartbeat connection when the Virtual Chassis member routers reside in the same subnet, see "Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in the Same Subnet" on page 259.

Requirements

This example uses the following software and hardware components:

- Junos OS Release 14.1 and later releases
- One MX240 Universal Routing Platform
- One MX480 Universal Routing Platform

This configuration example has been tested using the software release listed and is assumed to work on all later releases.

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.

Before you configure a heartbeat connection for a Virtual Chassis:

- Configure a Virtual Chassis consisting of two MX Series routers.

  See "Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis" on page 69

  As part of the preprovisioned Virtual Chassis configuration shown in the configuration example, you must create and apply the `member0-re0`, `member0-re1`, `member1-re0`, and `member1-re1` configuration groups for each member Routing Engine. Each configuration group includes a unique IP address for the management Ethernet interface (`fxp0`) on each Routing Engine.

  NOTE: When you create the preprovisioned Virtual Chassis configuration at the `[edit virtual-chassis]` hierarchy level, make sure you do *not* configure the `no-split-detection` statement to disable detection of a split in the Virtual Chassis. Using the `no-split-detection` statement is prohibited when you configure a Virtual Chassis heartbeat connection, and doing so causes the commit operation to fail.

- Ensure TCP connectivity between the primary Routing Engine in the Virtual Chassis primary router (VC-Pp) and the primary Routing Engine in the Virtual Chassis backup router (VC-Bp).

  The Virtual Chassis heartbeat connection opens a proprietary TCP port numbered 33087 on the VC-Pp to listen for heartbeat messages. If your network design includes firewalls or filters, make sure the network allows traffic between TCP port 33087 on the VC-Pp and the dynamically allocated TCP port on the VC-Bp.

Overview

IN THIS SECTION

A *heartbeat connection* is an IP-based, bidirectional packet connection between the primary router and backup router in an MX Series Virtual Chassis. The member routers forming the heartbeat connection exchange *heartbeat packets* that provide critical information about the availability and health of each member router. During a disruption or split in the Virtual Chassis configuration, the heartbeat connection prevents the member routers from changing primary role roles unnecessarily, which can cause undesirable results.

This example configures a heartbeat connection for an MX Series Virtual Chassis in which the two member routers, each with dual Routing Engines installed, reside in different subnets. Member router `gladius` resides in subnet 10.4.0.0/16 and is the global primary router for the Virtual Chassis (VC-P). Member router `trefoil` resides in subnet 10.5.0.0/16 and is the global backup router (VC-B) for the Virtual Chassis. The heartbeat connection is configured between the primary Routing Engine in `gladius` (represented by VC-Pp or `member0-re0`) and the primary Routing Engine in `trefoil` (represented by VC-Bp or `member1-re0`).

Configuring a heartbeat connection for an MX Series Virtual Chassis when the member routers reside in different subnets consists of the following tasks:

1. Configure two `master-only` IP addresses for the `fxp0` management interface: one for the member routers in subnet 10.4.0.0, and a different address for the member routers in subnet 10.5.0.0.

2. Configure a network path for the heartbeat connection to ensure that both member routers can reach each other's networks.

   This example creates static routes to both subnet 10.4.0.0 and subnet 10.5.0.0 on each member router.

3. Configure the Virtual Chassis heartbeat address for each member Routing Engine to cross-connect to the `master-only` IP address for the corresponding member Routing Engine in the other subnet.

4. (Optional) Configure a nondefault value for the Virtual Chassis heartbeat timeout interval.

To establish the heartbeat connection in a two-member MX Series Virtual Chassis, you must configure the heartbeat address to establish the connection between the primary and backup member routers. To ensure consistent access to the primary Routing Engine in the Virtual Chassis primary router (VC-Pp) regardless of which Routing Engine is currently active, you set the heartbeat address to the previously configured `master-only` IP address for the `fxp0` management interface.

Because the Virtual Chassis member routers in this example are in different subnets, you must configure a heartbeat address for each Routing Engine to enable a cross-connection to the `master-only` IP address for the corresponding Routing Engine in the other subnet, as shown in Table 22 on page 277:

Table 22: Heartbeat Cross-Connections for Member Routers in Different Subnets

Routing Engine	Subnet	Cross-connected Routing Engine	Heartbeat Address
`member0-re0`	10.4.0.0/16	`member1-re0`	10.5.2.210
`member0-re1`	10.4.0.0/16	`member1-re1`	10.5.2.210
`member1-re0`	10.5.0.0/16	`member0-re0`	10.4.2.210
`member1-re1`	10.5.0.0/16	`member0-re1`	10.4.2.210

Topology

This example configures a heartbeat connection for an MX Series Virtual Chassis with member routers residing in different subnets. For redundancy, each member router is configured with two Virtual Chassis ports.

Table 23 on page 278 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 23: Components of the Sample MX Series Virtual Chassis with Member Routers in Different Subnets

Router Name	Hardware	Serial Number	Member ID	Role	Virtual Chassis Ports	Subnet
gladius	MX240 router with: • Primary 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 (primary)	vcp-2/2/0 vcp-2/3/0	10.4.0.0/16
trefoil	MX480 router with: • Primary 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	10.5.0.0/16

Configuration

IN THIS SECTION

To configure a heartbeat connection in an MX Series Virtual Chassis with member routers in different subnets, perform these tasks:

CLI Quick Configuration

To quickly configure a heartbeat connection for a Virtual Chassis with with member routers in different subnets, copy the following commands and paste them into the router terminal window:

```
[edit]
set groups member0-re0 interfaces fxp0 unit 0 family inet address 10.4.2.210/16 master-only
set groups member0-re1 interfaces fxp0 unit 0 family inet address 10.4.2.210/16 master-only
set groups member1-re0 interfaces fxp0 unit 0 family inet address 10.5.2.210/16 master-only
set groups member1-re1 interfaces fxp0 unit 0 family inet address 10.5.2.210/16 master-only
set groups member0-re0 routing-options static route 10.4.0.0/16 next-hop 10.4.0.1
set groups member0-re0 routing-options static route 10.4.0.0/16 retain
set groups member0-re0 routing-options static route 10.4.0.0/16 no-readvertise
set groups member0-re0 routing-options static route 10.5.0.0/16 next-hop 10.4.0.1
set groups member0-re0 routing-options static route 10.5.0.0/16 retain
set groups member0-re0 routing-options static route 10.5.0.0/16 no-readvertise
set groups member0-re1 routing-options static route 10.4.0.0/16 next-hop 10.4.0.1
set groups member0-re1 routing-options static route 10.4.0.0/16 retain
set groups member0-re1 routing-options static route 10.4.0.0/16 no-readvertise
set groups member0-re1 routing-options static route 10.5.0.0/16 next-hop 10.4.0.1
set groups member0-re1 routing-options static route 10.5.0.0/16 retain
set groups member0-re1 routing-options static route 10.5.0.0/16 no-readvertise
set groups member1-re0 routing-options static route 10.5.0.0/16 next-hop 10.5.0.1
set groups member1-re0 routing-options static route 10.5.0.0/16 retain
set groups member1-re0 routing-options static route 10.5.0.0/16 no-readvertise
```

```
set groups member1-re0 routing-options static route 10.4.0.0/16 next-hop 10.5.0.1
set groups member1-re0 routing-options static route 10.4.0.0/16 retain
set groups member1-re0 routing-options static route 10.4.0.0/16 no-readvertise
set groups member1-re1 routing-options static route 10.5.0.0/16 next-hop 10.5.0.1
set groups member1-re1 routing-options static route 10.5.0.0/16 retain
set groups member1-re1 routing-options static route 10.5.0.0/16 no-readvertise
set groups member1-re1 routing-options static route 10.4.0.0/16 next-hop 10.5.0.1
set groups member1-re1 routing-options static route 10.4.0.0/16 retain
set groups member1-re1 routing-options static route 10.4.0.0/16 no-readvertise
set groups member0-re0 virtual-chassis heartbeat-address 10.5.2.210
set groups member0-re1 virtual-chassis heartbeat-address 10.5.2.210
set groups member1-re0 virtual-chassis heartbeat-address 10.4.2.210
set groups member1-re1 virtual-chassis heartbeat-address 10.4.2.210
set virtual-chassis heartbeat-timeout 10
```

Configuring a Consistent Management IP Address for Each Routing Engine

Step-by-Step Procedure

In addition to configuring a unique IP address for the `fxp0` management interface on each Routing Engine when you first set up the Virtual Chassis, you must configure additional management IP addresses, known as the `master-only` address, to ensure consistent access to the `fxp0` management interface on the primary Routing Engine in the Virtual Chassis primary router (VC-Pp). The `master-only` address is active only on the management interface for the VC-Pp. During a switchover, the `master-only` address moves to the new Routing Engine currently functioning as the VC-Pp.

Because the Virtual Chassis primary router and backup router in this example reside in different subnets, you must configure two different `master-only` IP addresses: one for the Routing Engines in subnet 10.4.0.0/16 (`member0-re0` and `member0-re1`), and one for the Routing Engines in subnet 10.5.0.0/16 (`member1-re0` and `member1-re1`). You then configure these `master-only` addresses as the subnet-specific heartbeat addresses to establish the heartbeat connection. For more information about the cross-connections in this example, see Table 22 on page 277.

To configure the primary-only `fxp0` IP address for each Routing Engine:

1. From the console on member 0, configure the IP address for the `fxp0` management interface for the Routing Engines in subnet 10.4.0.0/16.

```
{master:member0-re0}[edit]
user@gladius# set groups member0-re0 interfaces fxp0 unit 0 family inet address 10.4.2.210/16
primary-only
```

```
user@gladius# set groups member0-re1 interfaces fxp0 unit 0 family inet address 10.4.2.210/16
master-only
```

2. From the console on member 0, configure the IP address for the `fxp0` management interface for the Routing Engines in subnet 10.5.0.0/16.

```
{master:member0-re0}[edit]
user@gladius# set groups member1-re0 interfaces fxp0 unit 0 family inet address 10.5.2.210/16
primary-only
user@gladius# set groups member1-re1 interfaces fxp0 unit 0 family inet address 10.5.2.210/16
master-only
```

Results

From the console on the Virtual Chassis primary router, display the results of the configuration. For brevity, portions of the configuration unrelated to this procedure are replaced by an ellipsis (...).

For `member0-re0`:

```
{master:member0-re0}[edit]
user@gladius# show groups member0-re0
system {
    host-name gladius;
    backup-router 10.4.0.1 destination [ 172.16.0.0/12 ... 10.204.0.0/16 ];
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
                address 10.4.2.100/16;
                address 10.4.2.210/16 {
                    master-only;
                }
            }
        }
    }
}
```

For `member0-re1`:

```
{master:member0-re0}[edit]
user@gladius# show groups member0-re1
system {
    host-name gladius1;
    backup-router 10.4.0.1 destination [ 172.16.0.0/12 ... 10.204.0.0/16 ];
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
                address 10.4.2.101/16;
                address 10.4.2.210/16 {
                    master-only;
                }
            }
        }
    }
}
```

For `member1-re0`:

```
{master:member0-re0}[edit]
user@gladius# show groups member1-re0
system {
    host-name trefoil;
    backup-router 10.5.0.1 destination [ 172.16.0.0/12 ... 10.204.0.0/16 ];
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
                address 10.5.3.101/16;
                address 10.5.2.210/16 {
                    master-only;
                }
            }
        }
    }
}
```

For member1-re1:

```
{master:member0-re0}[edit]
user@gladius# show groups member1-re1
system {
    host-name trefoil1;
    backup-router 10.5.0.1 destination [ 172.16.0.0/12 ... 10.204.0.0/16 ];
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
                address 10.5.3.102/16;
                address 10.5.2.210/16 {
                    master-only;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter commit from configuration mode.

Configuring Static Routes for Both Subnets on Each Routing Engine

Step-by-Step Procedure

You must configure secure and reliable routes for subnets 10.4.0.0/16 and 10.5.0.0/16 on each Routing Engine for the exchange of TCP/IP heartbeat packets. The heartbeat packets provide critical information about the availability and health of each member router.

The routes you configure for the heartbeat connection must be independent of the Virtual Chassis port links. Specifically, you must ensure that the primary Routing Engine in the Virtual Chassis backup router (VC-Bp) can make a TCP/IP connection to the master-only IP address of the primary Routing Engine in the Virtual Chassis primary router (VC-Pp).

This examples creates static routes to both subnets on each member Routing Engine to configure the heartbeat path. However, you can choose the method that best meets your needs to configure the heartbeat path for member routers in different subnets.

BEST PRACTICE: We recommend that you use the router management interface (`fxp0`) as the heartbeat path. The management interface is generally available earlier than the line card interfaces, and is typically connected to a more secure network than the other interfaces.

To create static routes for subnets 10.4.0.0/16 and 10.5.0.0/16 on each Routing Engine:

1. Log in to the console on member 0 (Virtual Chassis primary router).
2. Configure the static routes for `member0-re0`.

```
{master:member0-re0}[edit]
user@gladius# set groups member0-re0 routing-options static route 10.4.0.0/16 next-hop
10.4.0.1
user@gladius# set groups member0-re0 routing-options static route 10.4.0.0/16 retain
user@gladius# set groups member0-re0 routing-options static route 10.4.0.0/16 no-readvertise
user@gladius# set groups member0-re0 routing-options static route 10.5.0.0/16 next-hop
10.4.0.1
user@gladius# set groups member0-re0 routing-options static route 10.5.0.0/16 retain
user@gladius# set groups member0-re0 routing-options static route 10.5.0.0/16 no-readvertise
```

3. Configure the static routes for `member0-re1`.

```
{master:member0-re0}[edit]
user@gladius# set groups member0-re1 routing-options static route 10.4.0.0/16 next-hop
10.4.0.1
user@gladius# set groups member0-re1 routing-options static route 10.4.0.0/16 retain
user@gladius# set groups member0-re1 routing-options static route 10.4.0.0/16 no-readvertise
user@gladius# set groups member0-re1 routing-options static route 10.5.0.0/16 next-hop
10.4.0.1
user@gladius# set groups member0-re1 routing-options static route 10.5.0.0/16 retain
user@gladius# set groups member0-re1 routing-options static route 10.5.0.0/16 no-readvertise
```

4. Configure the static routes for `member1-re0`.

```
{master:member0-re0}[edit]
user@gladius# set groups member1-re0 routing-options static route 10.5.0.0/16 next-hop
10.5.0.1
user@gladius# set groups member1-re0 routing-options static route 10.5.0.0/16 retain
```

```
user@gladius# set groups member1-re0 routing-options static route 10.5.0.0/16 no-readvertise
user@gladius# set groups member1-re0 routing-options static route 10.4.0.0/16 next-hop
10.5.0.1
user@gladius# set groups member1-re0 routing-options static route 10.4.0.0/16 retain
user@gladius# set groups member1-re0 routing-options static route 10.4.0.0/16 no-readvertise
```

5. Configure the static routes for `member1-re1`.

```
{master:member0-re0}[edit]
user@gladius# set groups member1-re1 routing-options static route 10.5.0.0/16 next-hop
10.5.0.1
user@gladius# set groups member1-re1 routing-options static route 10.5.0.0/16 retain
user@gladius# set groups member1-re1 routing-options static route 10.5.0.0/16 no-readvertise
user@gladius# set groups member1-re1 routing-options static route 10.4.0.0/16 next-hop
10.5.0.1
user@gladius# set groups member1-re1 routing-options static route 10.4.0.0/16 retain
user@gladius# set groups member1-re1 routing-options static route 10.4.0.0/16 no-readvertise
```

Results

Display the results of the configuration. For brevity, portions of the configuration unrelated to this procedure are replaced by an ellipsis (...).

For `member0-re0`:

```
{master:member0-re0}[edit]
user@gladius# show groups member0-re0 routing-options static
route 10.4.0.0/16 {
    next-hop 10.4.0.1;
    retain;
    no-readvertise;
}
route 10.5.0.0/16 {
    next-hop 10.4.0.1;
    retain;
    no-readvertise;
}
...
```

For `member0-re1`:

```
{master:member0-re0}[edit]
user@gladius# show groups member0-re1 routing-options static
route 10.4.0.0/16 {
    next-hop 10.4.0.1;
    retain;
    no-readvertise;
}
route 10.5.0.0/16 {
    next-hop 10.4.0.1;
    retain;
    no-readvertise;
}
...
```

For `member1-re0`:

```
{master:member0-re0}[edit]
user@gladius# show groups member1-re0 routing-options static
route 10.5.0.0/16 {
    next-hop 10.5.0.1;
    retain;
    no-readvertise;
}
route 10.4.0.0/16 {
    next-hop 10.5.0.1;
    retain;
    no-readvertise;
}
...
```

For `member1-re1`:

```
{master:member0-re0}[edit]
user@gladius# show groups member1-re1 routing-options static
route 10.5.0.0/16 {
    next-hop 10.5.0.1;
    retain;
    no-readvertise;
}
```

```
route 10.4.0.0/16 {
    next-hop 10.5.0.1;
    retain;
    no-readvertise;
}
...
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Heartbeat Address and Heartbeat Timeout

Step-by-Step Procedure

To enable cross-connection between Virtual Chassis member routers in different subnets, you configure 10.5.2.210, which is the `master-only` IP address for the Routing Engines in subnet 10.5.0.0/16, as the heartbeat address for the Routing Engines in subnet 10.4.0.0/16 (`member0-re0` and `member0-re1`). Conversely, you configure 10.4.2.210, which is the `master-only` IP address for the Routing Engines in subnet 10.4.0.0/16, as the heartbeat address for the Routing Engines in subnet 10.5.0.0/16 (`member1-re0` and `member1-re1`). For more information about the cross-connections in this example, see Table 22 on page 277.

Optionally, you can also configure a nondefault value for the heartbeat timeout interval. The heartbeat timeout is the maximum time within which a Virtual Chassis member router must respond to a heartbeat packet sent by the other member router. If you do not explicitly configure the heartbeat timeout interval, the default value (2 seconds) applies.

To configure the heartbeat address and heartbeat timeout:

1. Log in to the console on member 0 (Virtual Chassis primary router).

2. Configure the heartbeat address for each Routing Engine.

```
{master:member0-re0}[edit]
user@gladius# set groups member0-re0 virtual-chassis heartbeat-address 10.5.2.210
user@gladius# set groups member0-re1 virtual-chassis heartbeat-address 10.5.2.210
user@gladius# set groups member1-re0 virtual-chassis heartbeat-address 10.4.2.210
user@gladius# set groups member1-re1 virtual-chassis heartbeat-address 10.4.2.210
```

3. (Optional) Configure a nondefault value for the heartbeat timeout interval.

```
{master:member0-re0}[edit]
user@gladius# set virtual-chassis heartbeat-timeout 10
```

Results

Display the results of the configuration.

For `member0-re0`:

```
{master:member0-re0}[edit]
user@gladius# show groups member0-re0 virtual-chassis
preprovisioned;
traceoptions {
    file VCCP size 100m;
    flag all;
}
heartbeat-address 10.5.2.210;
heartbeat-timeout 10;
member 0 {
    role routing-engine;
    serial-number JN10C7135AFC;
}
member 1 {
    role routing-engine;
    serial-number JN115D117AFB;
}
```

For `member0-re1`:

```
{master:member0-re0}[edit]
user@gladius# show groups member0-re1 virtual-chassis
preprovisioned;
traceoptions {
    file VCCP size 100m;
    flag all;
}
heartbeat-address 10.5.2.210;
heartbeat-timeout 10;
```

```
member 0 {
    role routing-engine;
    serial-number JN10C7135AFC;
}
member 1 {
    role routing-engine;
    serial-number JN115D117AFB;
}
```

For `member1-re0`:

```
{master:member0-re0}[edit]
user@gladius# show groups member1-re0 virtual-chassis
preprovisioned;
traceoptions {
    file VCCP size 100m;
    flag all;
}
heartbeat-address 10.4.2.210;
heartbeat-timeout 10;
member 0 {
    role routing-engine;
    serial-number JN10C7135AFC;
}
member 1 {
    role routing-engine;
    serial-number JN115D117AFB;
}
```

For `member1-re1`:

```
{master:member0-re0}[edit]
user@gladius# show groups member1-re1 virtual-chassis
preprovisioned;
traceoptions {
    file VCCP size 100m;
    flag all;
}
heartbeat-address 10.4.2.210;
heartbeat-timeout 10;
member 0 {
```

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

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

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

Verifying the Virtual Chassis Heartbeat Connection

Purpose

Verify that the heartbeat connection between the Virtual Chassis member routers is properly configured and operational.

Action

Display the state of one or both member routers when a heartbeat connection is configured.

```
{master:member0-re0}
user@gladius> show virtual-chassis heartbeat
member0:
---------------------------------------------------------------------------
```

```
Local         Remote          State        Time
10.4.2.210    10.5.3.101      Alive        2014-03-18 10:18:14 PST

member1:
--------------------------------------------------------------------------
Local         Remote          State        Time
10.5.3.101    10.4.2.210      Alive        2014-03-18 10:18:15 PST
```

Meaning

For each member router, the command output displays the IP addresses of the local and remote member routers that form the heartbeat connection. The value `Alive` in the `State` field confirms that the primary Routing Engine in the specified member router is connected and has received a heartbeat response message. The `Time` field specifies the date and time of the last connection state change.

Verifying Use of the Heartbeat Connection During an Adjacency Split or Disruption

Purpose

Verify use of the heartbeat connection when an adjacency disruption or split is detected in the Virtual Chassis.

Action

Display the status of the member routers in the Virtual Chassis:

```
{master:member0-re0}
user@gladius> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: a5b6.be0c.9525

                                               Mastership          Neighbor List
Member ID        Status   Serial No    Model    priority    Role      ID  Interface

0 (FPC   0- 11) Heartbt  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 `Status` field for member ID 0 displays `Heartbt`, which indicates that this member router has used the heartbeat packet connection to maintain primary role roles during an adjacency disruption or split in the Virtual Chassis configuration. The `Status` field for member ID 1 displays `Prsnt`, which indicates that this member router is connected to the Virtual Chassis.

If a router is not currently connected to the Virtual Chassis, the `Status` field displays `NotPrsnt`.

Verifying Virtual Chassis Member Health from Heartbeat Statistics

Purpose

Use statistics collected by the heartbeat connection to verify the availability and health of each Virtual Chassis member router. You can also use the `show virtual-chassis heartbeat detail` command to determine the maximum latency and minimum latency in your network.

Action

Display and review the statistics collected by the heartbeat connection.

```
{master:member0-re0}
user@gladius> show virtual-chassis heartbeat detail
member0:
--------------------------------------------------------------------------
Local          Remote         State       Time
10.4.2.210     10.5.3.101     Alive       2014-03-18 10:18:14 PST

Heartbeat statistics
  Heartbeats sent: 10079
  Heartbeats received: 10079
  Heartbeats lost/missed: 0
  Last time sent: 2014-03-18 20:03:10 PST (00:00:00 ago)
  Last time received: 2014-03-18 20:03:10 PST (00:00:00 ago)
  Maximum latency (secs): 0
  Minimum latency (secs): 0

member1:
--------------------------------------------------------------------------
Local          Remote         State       Time
10.5.3.101     10.4.2.210     Alive       2014-03-18 10:18:15 PST
```

```
Heartbeat statistics
  Heartbeats sent: 10083
  Heartbeats received: 10083
  Heartbeats lost/missed: 0
  Last time sent: 2014-02-18 20:03:09 PST (00:00:01 ago)
  Last time received: 2014-02-18 20:03:09 PST (00:00:01 ago)
  Maximum latency (secs): 0
  Minimum latency (secs): 0
```

Meaning

In this example, the number of heartbeat request messages sent (`Heartbeats sent`) equals the number of heartbeat response messages received (`Heartbeats received`), with no heartbeat messages lost (`Heartbeats lost/missed`). This indicates that both member routers forming the heartbeat connection are available and operational. Any difference between `Heartbeats sent` and `Heartbeats received` appears in the `Heartbeats lost/missed` field.

The `Maximum latency` and `Minimum latency` fields measure the maximum and minimum number of seconds that elapse on the local router between transmission of a heartbeat request message and receipt of a heartbeat response message. In this example, the value `0` in the `Maximum latency` and `Minimum latency` fields indicates that there is no measurable network delay caused by this operation. You can use the `Maximum latency` value to determine whether you need to increase the `heartbeat-timeout` to a value higher than the default (2 seconds). If the maximum latency in your network is too high to accommodate a 2-second `heartbeat-timeout` value, increasing the `heartbeat-timeout` interval enables you to account for network delay when a Virtual Chassis adjacency disruption or split occurs.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can configure a heartbeat connection in an MX Series Virtual Chassis.

RELATED DOCUMENTATION

Example: Determining Member Health Using an MX Series Virtual Chassis Heartbeat Connection with Member Routers in the Same Subnet | 259

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring a Virtual Chassis Heartbeat Connection | 17
Global Roles and Local Roles in a Virtual Chassis | 14

Configuring a Consistent Management IP Address

12

CHAPTER

Tracing Virtual Chassis Operations for Troubleshooting Purposes

IN THIS CHAPTER

Tracing Virtual Chassis Operations for MX Series 5G Universal Routing Platforms

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 primary (also referred to as the *protocol primary*) 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 System Log Explorer.)

   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 297.
2. (Optional) Configure characteristics of the trace log file.

   See "Configuring Characteristics of the Virtual Chassis Trace Log File" on page 297.
3. (Optional) Configure user access to the trace log file.

   See "Configuring Access to the Virtual Chassis Trace Log File" on page 299.
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 300.
5. Configure flags to specify the Virtual Chassis operations that you want to trace.

   See "Configuring the Virtual Chassis Operations to Trace" on page 300.

RELATED DOCUMENTATION

Configuring Preprovisioned Member Information for a Virtual Chassis | 56
Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring the Name of the Virtual Chassis Trace Log File

To trace operations for a 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 Virtual Chassis operations:

- On the device to be designated as the primary 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 5G Universal Routing Platforms | 296
Configuring Preprovisioned Member Information for a Virtual Chassis | 56
Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring Characteristics of the Virtual Chassis Trace Log File

You can optionally configure the following characteristics of the trace log file for a 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 or switch appends new information to an existing trace file. You can optionally specify that the router or switch replace an exisiting trace file instead of appending information to it.

To configure the maximum number and maximum size of trace files:

- On the router or switch to be designated as the primary of the 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 or switch from the placing a timestamp on the trace log file:

- On the router or switch to be designated as the primary of the 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 exisiting trace file instead of appending information to it:

- On the router or switch to be designated as the primary of the Virtual Chassis, specify that the router or switch replaces an existing trace file:

```
[edit virtual-chassis]
user@host# set traceoptions file filename replace
```

RELATED DOCUMENTATION

Tracing Virtual Chassis Operations for MX Series 5G Universal Routing Platforms | 296
Configuring Preprovisioned Member Information for a Virtual Chassis | 56
Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

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 or switch to be designated as the primary 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 or switch to be designated as the primary 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 5G Universal Routing Platforms | 296

Configuring Preprovisioned Member Information for a Virtual Chassis | 56

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

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 a Virtual Chassis by including regular expressions to be matched.

To refine the output of the trace log file:

- On the router or switch to be designated as the primary 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 5G Universal Routing Platforms | 296

Configuring Preprovisioned Member Information for a Virtual Chassis | 56

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

Configuring the Virtual Chassis Operations to Trace

By default, the router or switch logs only important events. You can specify which operations to trace for a Virtual Chassis by including specific tracing flags when you configure tracing. Table 24 on page 301 describes the flags that you can include.

Table 24: Tracing Flags for 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 primary-role 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.

Table 24: Tracing Flags for Virtual Chassis *(Continued)*

Flag	Description
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 primary-role 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 5G Universal Routing Platforms | 296

Configuring Preprovisioned Member Information for a Virtual Chassis | 56

Example: Configuring Interchassis Redundancy for MX Series 5G Universal Routing Platforms Using a Virtual Chassis | 69

13

CHAPTER

Configuration Statements and Operational Commands

IN THIS CHAPTER

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements
- Operational Commands