Junos® OS

High Availability Configuration Guide

Release
12.3

Published: 2013-01-02

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

Junos® OS High Availability Configuration Guide
Release 12.3

Revision History
October 2012—R1 Junos OS 12.3

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Abbreviated Table of Contents

Table of Contents

List of Figures

List of Tables

About This Guide

This preface provides the following guidelines for using the *Junos® OS High Availability Configuration Guide*:

Junos OS Documentation and Release Notes

For a list of related Junos OS documentation, see http://www.juniper.net/techpubs/software/junos/.

If the information in the latest release notes differs from the information in the documentation, follow the *Junos OS Release Notes.*

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

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using the Junos operating system (Junos OS) and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using Junos OS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at http://www.juniper.net/books.

Objectives

This guide is designed to provide an overview of high availability concepts and techniques. By understanding the redundancy features of Juniper Networks routing platforms and Junos OS, a network administrator can enhance the reliability of a network and deliver highly available services to customers.

NOTE: For additional information about the Junos OS—either corrections to or information that might have been omitted from this guide—see the software release notes at http://www.juniper.net/ .

Audience

This guide is designed for network administrators who are configuring and monitoring a Juniper Networks M Series, MX Series, T Series, EX Series, or J Series router or switch.

To use this guide, you need a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. You must also be familiar with one or more of the following Internet routing protocols:

- Border Gateway Protocol (BGP)
- Distance Vector Multicast Routing Protocol (DVMRP)
- Intermediate System-to-Intermediate System (IS-IS)
- Internet Control Message Protocol (ICMP) router discovery
- Internet Group Management Protocol (IGMP)
- Multiprotocol Label Switching (MPLS)
- Open Shortest Path First (OSPF)
- Protocol-Independent Multicast (PIM)
- Resource Reservation Protocol (RSVP)
- Routing Information Protocol (RIP)
- Simple Network Management Protocol (SNMP)

Personnel operating the equipment must be trained and competent; must not conduct themselves in a careless, willfully negligent, or hostile manner; and must abide by the instructions provided by the documentation.

Supported Platforms

For the features described in this manual, the Junos OS currently supports the following platforms:

- J Series
- M Series
- MX Series
- SRX Series
- T Series
- EX Series

Using the Indexes

This reference contains two indexes: a complete index that includes topic entries, and an index of statements and commands only.

In the index of statements and commands, an entry refers to a statement summary section only. In the complete index, the entry for a configuration statement or command contains at least two parts:

- The primary entry refers to the statement summary section.
- The secondary entry, *usage guidelines,* refers to the section in a configuration guidelines chapter that describes how to use the statement or command.

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the**configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*

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

<table>
<tr><th>Convention</th><th>Description</th><th>Examples</th></tr>
<tr><td>Text like this</td><td>Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.</td><td>• To configure a stub area, include the stub statement at the[edit protocols ospf area area-id] hierarchy level.
• The console port is labeled CONSOLE.</td></tr>
<tr><td>< > (angle brackets)</td><td>Enclose optional keywords or variables.</td><td>stub <default-metric <i>metric</i>>;</td></tr>
<tr><td>| (pipe symbol)</td><td>Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.</td><td>broadcast | multicast

(string1 | string2 | string3)</td></tr>
<tr><td># (pound sign)</td><td>Indicates a comment specified on the same line as the configuration statement to which it applies.</td><td>rsvp { # Required for dynamic MPLS only</td></tr>
<tr><td>[ ] (square brackets)</td><td>Enclose a variable for which you can substitute one or more values.</td><td>community name members [community-ids]</td></tr>
<tr><td>Indention and braces ({ })</td><td>Identify a level in the configuration hierarchy.</td><td rowspan="2"><pre>[edit]
routing-options {
  static {
    route default {
      nexthop address;
      retain;
    }
  }
}</pre></td></tr>
<tr><td>; (semicolon)</td><td>Identifies a leaf statement at a configuration hierarchy level.</td></tr>
<tr><td colspan="3">J-Web GUI Conventions</td></tr>
<tr><td>Bold text like this</td><td>Represents J-Web graphical user interface (GUI) items you click or select.</td><td>• In the Logical Interfaces box, select All Interfaces.
• To cancel the configuration, click Cancel.</td></tr>
<tr><td>> (bold right angle bracket)</td><td>Separates levels in a hierarchy of J-Web selections.</td><td>In the configuration editor hierarchy, select Protocols>Ospf.</td></tr>
</table>

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number

- Software release version (if applicable)

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

CHAPTER 1

High Availability Overview

This chapter contains the following topics:

Understanding High Availability Features on Juniper Networks Routers

For Juniper Networks routing platforms running the Junos® operating system (Junos OS), *high availability* refers to the hardware and software components that provide redundancy and reliability for packet-based communications. This topic provides brief overviews of the following high availability features:

Routing Engine Redundancy

Redundant Routing Engines are two Routing Engines that are installed in the same routing platform. One functions as the master, while the other stands by as a backup should the master Routing Engine fail. On routing platforms with dual Routing Engines, network reconvergence takes place more quickly than on routing platforms with a single Routing Engine.

Graceful Routing Engine Switchover

Graceful Routing Engine switchover (GRES) enables a routing platform with redundant Routing Engines to continue forwarding packets, even if one Routing Engine fails. Graceful

Routing Engine switchover preserves interface and kernel information. Traffic is not interrupted. However, graceful Routing Engine switchover does not preserve the control plane. Neighboring routers detect that the router has experienced a restart and react to the event in a manner prescribed by individual routing protocol specifications. To preserve routing during a switchover, graceful Routing Engine switchover must be combined with either graceful restart protocol extensions or nonstop active routing. For more information, see "Understanding Graceful Routing Engine Switchover in the Junos OS" on page 51.

Nonstop Bridging

Nonstop bridging enables a routing platform with redundant Routing Engines to switch from a primary Routing Engine to a backup Routing Engine without losing Layer 2 Control Protocol (L2CP) information. Nonstop bridging uses the same infrastructure as graceful Routing Engine switchover to preserve interface and kernel information. However, nonstop bridging also saves L2CP information by running the Layer 2 Control Protocol process (l2cpd) on the backup Routing Engine.

NOTE: To use nonstop bridging, you must first enable graceful Routing Engine switchover.

Nonstop bridging is supported for the following Layer 2 control protocols:

- Spanning Tree Protocol (STP)
- Rapid Spanning Tree Protocol (RSTP)
- Multiple Spanning Tree Protocol (MSTP)

For more information, see "Nonstop Bridging Concepts" on page 67.

Nonstop Active Routing

Nonstop active routing (NSR) enables a routing platform with redundant Routing Engines to switch from a primary Routing Engine to a backup Routing Engine without alerting peer nodes that a change has occurred. Nonstop active routing uses the same infrastructure as graceful Routing Engine switchover to preserve interface and kernel information. However, nonstop active routing also preserves routing information and protocol sessions by running the routing protocol process (rpd) on both Routing Engines. In addition, nonstop active routing preserves TCP connections maintained in the kernel.

NOTE: To use nonstop active routing, you must also configure graceful Routing Engine switchover.

For a list of protocols and features supported by nonstop active routing, see "Nonstop Active Routing Protocol and Feature Support" on page 81.

For more information about nonstop active routing, see "Nonstop Active Routing Concepts" on page 77.

Graceful Restart

With routing protocols, any service interruption requires an affected router to recalculate adjacencies with neighboring routers, restore routing table entries, and update other protocol-specific information. An unprotected restart of a router can result in forwarding delays, route flapping, wait times stemming from protocol reconvergence, and even dropped packets. To alleviate this situation, graceful restart provides extensions to routing protocols. These protocol extensions define two roles for a router—*restarting* and *helper*. The extensions signal neighboring routers about a router undergoing a restart and prevent the neighbors from propagating the change in state to the network during a graceful restart wait interval. The main benefits of graceful restart are uninterrupted packet forwarding and temporary suppression of all routing protocol updates. Graceful restart enables a router to pass through intermediate convergence states that are hidden from the rest of the network.

When a router is running graceful restart and the router stops sending and replying to protocol liveness messages (hellos), the adjacencies assume a graceful restart and begin running a timer to monitor the restarting router. During this interval, helper routers do not process an adjacency change for the router that they assume is restarting, but continue active routing with the rest of the network. The helper routers assume that the router can continue stateful forwarding based on the last preserved routing state during the restart.

If the router was actually restarting and is back up before the graceful timer period expires in all of the helper routers, the helper routers provide the router with the routing table, topology table, or label table (depending on the protocol), exit the graceful period, and return to normal network routing.

If the router does not complete its negotiation with helper routers before the graceful timer period expires in all of the helper routers, the helper routers process the router's change in state and send routing updates, so that convergence occurs across the network. If a helper router detects a link failure from the router, the topology change causes the helper router to exit the graceful wait period and to send routing updates, so that network convergence occurs.

To enable a router to undergo a graceful restart, you must include the **graceful-restart** statement at either the global **[edit routing-options]** hierarchy level. When a routing session is started, a router that is configured with graceful restart must negotiate with its neighbors to support it when it undergoes a graceful restart. A neighboring router will accept the negotiation and support helper mode without requiring graceful restart to be configured on the neighboring router.

NOTE: A Routing Engine switchover event on a helper router that is in graceful wait state causes the router to drop the wait state and to propagate the adjacency's state change to the network.

Graceful restart is supported for the following protocols and applications:

- BGP
- ES-IS
- IS-IS
- OSPF/OSPFv3
- PIM sparse mode
- RIP/RIPng
- MPLS-related protocols, including:
  - Label Distribution Protocol (LDP)
  - Resource Reservation Protocol (RSVP)
  - Circuit cross-connect (CCC)
  - Translational cross-connect (TCC)
- Layer 2 and Layer 3 virtual private networks (VPNs)

For more information, see “Graceful Restart Concepts” on page 105.

Nonstop Active Routing Versus Graceful Restart

Nonstop active routing and graceful restart are two different methods of maintaining high availability. Graceful restart requires a restart process. A router undergoing a graceful restart relies on its neighbors (or helpers) to restore its routing protocol information. The restart is the mechanism by which helpers are signaled to exit the wait interval and start providing routing information to the restarting router.

In contrast, nonstop active routing does not involve a router restart. Both the master and standby Routing Engines are running the routing protocol process (rpd) and exchanging updates with neighbors. When one Routing Engine fails, the router simply switches to the active Routing Engine to exchange routing information with neighbors. Because of these feature differences, nonstop routing and graceful restart are mutually exclusive. Nonstop active routing cannot be enabled when the router is configured as a graceful restarting router. If you include the **graceful-restart** statement at any hierarchy level and the **nonstop-routing** statement at the **[edit routing-options]** hierarchy level and try to commit the configuration, the commit request fails.

Effects of a Routing Engine Switchover

“Effects of a Routing Engine Switchover” on page 51 describes the effects of a Routing Engine switchover when no high availability features are enabled and when graceful Routing Engine switchover, graceful restart, and nonstop active routing features are enabled.

VRRP

For Ethernet, Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, and logical interfaces, you can configure the Virtual Router Redundancy Protocol (VRRP) or VRRP for IPv6.

VRRP enables hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts. The VRRP routing platforms share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routing platforms is the master (active) and the others are backups. If the master fails, one of the backup routers becomes the new master router, providing a virtual default routing platform and allowing traffic on the LAN to be routed without relying on a single routing platform. Using VRRP, a backup router can take over a failed default router within a few seconds. This is done with minimum VRRP traffic and without any interaction with the hosts.

Routers running VRRP dynamically elect master and backup routers. You can also force assignment of master and backup routers using priorities from 1 through 255, with 255 being the highest priority. In VRRP operation, the default master router sends advertisements to backup routers at regular intervals. The default interval is 1 second. If a backup router does not receive an advertisement for a set period, the backup router with the next highest priority takes over as master and begins forwarding packets.

For more information, see "Understanding VRRP" on page 175.

Unified ISSU

A unified in-service software upgrade (unified ISSU) enables you to upgrade between two different Junos OS Releases with no disruption on the control plane and with minimal disruption of traffic. Unified ISSU is only supported by dual Routing Engine platforms. In addition, graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) must be enabled.

With a unified ISSU, you can eliminate network downtime, reduce operating costs, and deliver higher services levels. For more information, see "Unified ISSU Concepts" on page 239.

Interchassis Redundancy for MX Series Routers Using Virtual Chassis

Interchassis redundancy is a high availability feature that can span equipment located across multiple geographies to prevent network outages and protect 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. As more high-priority voice and video traffic is carried on the network, interchassis redundancy has become a requirement for providing stateful redundancy on broadband subscriber management equipment such as broadband services routers, broadband network gateways, and broadband remote access servers. Interchassis redundancy support enables service providers to fulfill strict service-level agreements (SLAs) and avoid unplanned network outages to better meet the needs of their customers.

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

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

Starting with Junos OS Release 11.2, Virtual Chassis configurations are supported on MX240, MX480, and MX960 3D Universal Edge Routers with Trio MPC/MIC interfaces and dual Routing Engines. In addition, graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) must be enabled on both member routers in the Virtual Chassis.

Related Documentation

• High Availability-Related Features in Junos OS on page 8

High Availability-Related Features in Junos OS

Related redundancy and reliability features include:

- Redundant power supplies, host modules, host subsystems, and forwarding boards. For more information, see the Junos OS System Basics Configuration Guide and the *Junos OS Hardware Network Operations Guide*.
- Additional link-layer redundancy, including Automatic Protection Switching (APS) for SONET interfaces, Multiplex Section Protection (MSP) for SDH interfaces, and DLSw redundancy for Ethernet interfaces. For more information, see the Junos® OS Network Interfaces.
- Bidirectional Forwarding Detection (BFD) works with other routing protocols to detect failures rapidly. For more information, see the Junos OS Routing Protocols Configuration Guide.
- Redirection of Multiprotocol Label Switching (MPLS) label-switched path (LSP) traffic—Mechanisms such as link protection, node-link protection, and fast reroute recognize link and node failures, allowing MPLS LSPs to select a bypass LSP to circumvent failed links or devices. For more information, see the Junos OS MPLS Applications Configuration Guide.

Related Documentation

• Understanding High Availability Features on Juniper Networks Routers on page 3

PART 2

Routing Engine and Switching Control Board Redundancy

CHAPTER 2

Routing Engine and Switching Control Board Redundancy Overview

For routers that have redundant Routing Engines or redundant switching control boards, including Switching and Forwarding Modules (SFMs). System and Switch Boards (SSBs), Forwarding Engine Boards (FEBs), or Compact Forwarding Engine Board (CFEBs), you can configure redundancy properties. This chapter includes the following topics:

Understanding Routing Engine Redundancy on Juniper Networks Routers

This topic contains the following sections:

Routing Engine Redundancy Overview

Redundant Routing Engines are two Routing Engines that are installed in the same routing platform. One functions as the master, while the other stands by as a backup should the master Routing Engine fail. On routing platforms with dual Routing Engines, network reconvergence takes place more quickly than on routing platforms with a single Routing Engine.

When a Routing Engine is configured as master, it has full functionality. It receives and transmits routing information, builds and maintains routing tables, communicates with interfaces and Packet Forwarding Engine components, and has full control over the chassis. When a Routing Engine is configured to be the backup, it does not communicate with the Packet Forwarding Engine or chassis components.

NOTE: On devices running Junos OS Release 8.4 or later, both Routing Engines cannot be configured to be master at the same time. This configuration causes the commit check to fail.

A failover from the master Routing Engine to the backup Routing Engine occurs automatically when the master Routing Engine experiences a hardware failure or when you have configured the software to support a change in mastership based on specific conditions. You can also manually switch Routing Engine mastership by issuing one of the **request chassis routing-engine** commands. In this topic, the term *failover* refers to an automatic event, whereas *switchover* refers to either an automatic or a manual event.

When a failover or a switchover occurs, the backup Routing Engine takes control of the system as the new master Routing Engine.

- If graceful Routing Engine switchover is not configured, when the backup Routing Engine becomes master, it resets the switch plane and downloads its own version of the microkernel to the Packet Forwarding Engine components. Traffic is interrupted while the Packet Forwarding Engine is reinitialized. All kernel and forwarding processes are restarted.
- If graceful Routing Engine switchover is configured, interface and kernel information is preserved. The switchover is faster because the Packet Forwarding Engines are not restarted. The new master Routing Engine restarts the routing protocol process (rpd). All hardware and interfaces are acquired by a process that is similar to a warm restart. For more information about graceful Routing Engine switchover, see "Understanding Graceful Routing Engine Switchover in the Junos OS" on page 51.
- If graceful Routing Engine switchover and nonstop active routing (NSR) are configured, traffic is not interrupted during the switchover. Interface, kernel, and routing protocol information is preserved. For more information about nonstop active routing, see "Nonstop Active Routing Concepts" on page 77.
- If graceful Routing Engine switchover and graceful restart are configured, traffic is not interrupted during the switchover. Interface and kernel information is preserved. Graceful restart protocol extensions quickly collect and restore routing information from the neighboring routers. For more information about graceful restart, see "Graceful Restart Concepts" on page 105.

Conditions That Trigger a Routing Engine Failover

The following events can result in an automatic change in Routing Engine mastership, depending on your configuration:

- The routing platform experiences a hardware failure. A change in Routing Engine mastership occurs if either the Routing Engine or the associated host module or subsystem is abruptly powered off. You can also configure the backup Routing Engine to take mastership if it detects a hard disk error on the master Routing Engine. To enable this feature, include the **failover on-disk-failure** statement at the **[edit chassis redundancy]** hierarchy level.

- The routing platform experiences a software failure, such as a kernel crash or a CPU lock. You must configure the backup Routing Engine to take mastership when it detects a loss of keepalive signal. To enable this failover method, include the **failover on-loss-of-keepalives** statement at the **[edit chassis redundancy]** hierarchy level.
- A specific software process fails. You can configure the backup Routing Engine to take mastership when one or more specified processes fail at least four times within 30 seconds. Include the **failover other-routing-engine** statement at the **[edit system processes *process-name*]** hierarchy level.

If any of these conditions is met, a message is logged and the backup Routing Engine attempts to take mastership. By default, an alarm is generated when the backup Routing Engine becomes active. After the backup Routing Engine takes mastership, it continues to function as master even after the originally configured master Routing Engine has successfully resumed operation. You must manually restore it to its previous backup status. (However, if at any time one of the Routing Engines is not present, the other Routing Engine becomes master automatically, regardless of how redundancy is configured.)

Default Routing Engine Redundancy Behavior

By default, Junos OS uses **re0** as the master Routing Engine and **re1** as the backup Routing Engine. Unless otherwise specified in the configuration, **re0** always becomes master when the acting master Routing Engine is rebooted.

NOTE: **A single Routing Engine in the chassis always becomes the master Routing Engine even if it was previously the backup Routing Engine.**

Perform the following steps to see how the default Routing Engine redundancy setting works:

1. Ensure that **re0** is the master Routing Engine.
2. Manually switch the state of Routing Engine mastership by issuing the **request chassis routing-engine master switch** command from the master Routing Engine. **re0** is now the backup Routing Engine and **re1** is the master Routing Engine.

   NOTE: **On the next reboot of the master Routing Engine, Junos OS returns the router to the default state because you have not configured the Routing Engines to maintain this state after a reboot.**

3. Reboot the master Routing Engine **re1**.

   The Routing Engine boots up and reads the configuration. Because you have not specified in the configuration which Routing Engine is the master, **re1** uses the default configuration as the backup. Now both **re0** and **re1** are in a backup state. Junos OS detects this conflict and, to prevent a no-master state, reverts to the default configuration to direct **re0** to become master.

Routing Engine Redundancy on a TX Matrix Router

On a routing matrix, all master Routing Engines in the TX Matrix router and connected T640 routers must run the same Junos OS Release. Likewise, all backup Routing Engines in a routing matrix must run the same Junos OS Release. When you run the same Junos OS Release on all master and backup Routing Engines in the routing matrix, a change in mastership to any backup Routing Engine in the routing matrix does not cause a change in mastership in any other chassis in the routing matrix.

If the same Junos OS Release is not running on all master and backup Routing Engines in the routing matrix, the following consequences occur when the **failover on-loss-of-keepalives** statement *is* included at the **[edit chassis redundancy]** hierarchy level:

- When the **failover on-loss-of-keepalives** statement is included at the **[edit chassis redundancy]** hierarchy level and you or a host subsystem initiates a change in mastership to the backup Routing Engine in the TX Matrix router, the master Routing Engines in the T640 routers detect a software release mismatch with the new master Routing Engine in the TX Matrix router and switch mastership to their backup Routing Engines.
- When you manually change mastership to a backup Routing Engine in a T640 router using the **request chassis routing-engine master** command, the new master Routing Engine in the T640 router detects a software release mismatch with the master Routing Engine in the TX Matrix router and relinquishes mastership to the original master Routing Engine. (Routing Engine mastership in the TX Matrix router does not switch in this case.)
- When a host subsystem initiates a change in mastership to a backup Routing Engine in a T640 router because the master Routing Engine has failed, the T640 router is logically disconnected from the TX Matrix router. To reconnect the T640 router, initiate a change in mastership to the backup Routing Engine in the TX Matrix router, or replace the failed Routing Engine in the T640 router and switch mastership to it. The replacement Routing Engine must be running the same software release as the master Routing Engine in the TX Matrix router.

If the same Junos OS Release is not running on all master and backup Routing Engines in the routing matrix, the following consequences occur when the **failover on-loss-of-keepalives** statement *is not* included at the **[edit chassis redundancy]** hierarchy level:

- If you initiate a change in mastership to the backup Routing Engine in the TX Matrix router, all T640 routers are logically disconnected from the TX Matrix router. To reconnect the T640 routers, switch mastership of all master Routing Engines in the T640 routers to their backup Routing Engines.
- If you initiate a change in mastership to a backup Routing Engine in a T640 router, the T640 router is logically disconnected from the TX Matrix router. To reconnect the T640 router, switch mastership of the new master Routing Engine in the T640 router back to the original master Routing Engine.

Situations That Require You to Halt Routing Engines

Before you shut the power off to a routing platform that has two Routing Engines or before you remove the master Routing Engine, you must first halt the backup Routing Engine and then halt the master Routing Engine. Otherwise, you might need to reinstall Junos OS. You can use the **request system halt both-routing-engines** command on the master Routing Engine, which first shuts down the master Routing Engine and then shuts down the backup Routing Engine. To shut down only the backup Routing Engine, issue the **request system halt** command on the backup Routing Engine.

If you halt the master Routing Engine and do not power it off or remove it, the backup Routing Engine remains inactive unless you have configured it to become the master when it detects a loss of keepalive signal from the master Routing Engine.

NOTE: To restart the router, you must log in to the console port (rather than the Ethernet management port) of the Routing Engine. When you log in to the console port of the master Routing Engine, the system automatically reboots. After you log in to the console port of the backup Routing Engine, press Enter to reboot it.

NOTE: If you have upgraded the backup Routing Engine, first reboot it and then reboot the master Routing Engine.

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Switching Control Board Redundancy on page 15
- Configuring Routing Engine Redundancy on page 25

Switching Control Board Redundancy

This section describes the following redundant switching control boards:

NOTE: A failover from a master switching control board to a backup switching control board occurs automatically when the master experiences a hardware failure or when you have configured the software to support a change in mastership based on specific conditions. You can also manually switch mastership by issuing specific **request chassis** commands. In this chapter, the term *failover* refers to an automatic event, whereas *switchover* refers to either an automatic or a manual event.

- Redundant CFEBs on the M10i Router on page 16
- Redundant FEBs on the M120 Router on page 16

Redundant CFEBs on the M10i Router

On the M10i router, the CFEB performs the following functions:

- Route lookups—Performs route lookups using the forwarding table stored in synchronous SRAM (SSRAM).
- Management of shared memory—Uniformly allocates incoming data packets throughout the router's shared memory.
- Transfer of outgoing data packets—Passes data packets to the destination Fixed Interface Card (FIC) or Physical Interface Card (PIC) when the data is ready to be transmitted.
- Transfer of exception and control packets—Passes exception packets to the microprocessor on the CFEB, which processes almost all of them. The remainder are sent to the Routing Engine for further processing. Any errors originating in the Packet Forwarding Engine and detected by the CFEB are sent to the Routing Engine using system log messages.

The M10i router has two CFEBs, one that is configured to act as the master and the other that serves as a backup in case the master fails. You can initiate a manual switchover by issuing the **request chassis cfeb master switch** command. For more information, see the Junos OS System Basics Configuration Guide.

Redundant FEBs on the M120 Router

The M120 router supports up to six Forwarding Engine Boards (FEBs). Flexible PIC Concentrator (FPCs), which host PICs, are separate from the FEBs, which handle packet forwarding. FPCs are located on the front of the chassis and provide power and management to PICs through the midplane. FEBs are located on the back of the chassis and receive signals from the midplane, which the FEBs process for packet forwarding. The midplane allows any FEB to carry traffic for any FPC.

To configure the mapping of FPCs to FEBs, use the **fpc-feb-connectivity** statement as described in the Junos OS System Basics Configuration Guide. You cannot specify a connection between an FPC and a FEB configured as a backup. If an FPC is not specified to connect to a FEB, the FPC is assigned automatically to the FEB with the same slot number. For example, the FPC in slot 1 is assigned to the FEB in slot 1.

You can configure one FEB as a backup for one or more FEBs by configuring a FEB redundancy group. When a FEB fails, the backup FEB can quickly take over packet forwarding. A redundancy group must contain exactly one backup FEB and can optionally contain one primary FEB and multiple other FEBs. A FEB can belong to only one group. A group can provide backup on a one-to-one basis (primary-to-backup), a many-to-one basis (two or more other-FEBs-to-backup), or a combination of both (one primary-to-backup and one or more other-FEBs-to-backup).

When you configure a primary FEB in a redundancy group, the backup FEB mirrors the exact forwarding state of the primary FEB. If switchover occurs from a primary FEB, the backup FEB does not reboot. A manual switchover from the primary FEB to the backup FEB results in less than 1 second of traffic loss. Failover from the primary FEB to the backup FEB results in less than 10 seconds of traffic loss.

If a failover occurs from the other FEB and a primary FEB is specified for the group, the backup FEB reboots so that the forwarding state from the other FEB can be downloaded to the backup FEB and forwarding can continue. Automatic failover from a FEB that is not specified as a primary FEB results in higher packet loss. The duration of packet loss depends on the number of interfaces and on the size of the routing table, but it can be minutes.

If a failover from a FEB occurs when no primary FEB is specified in the redundancy group, the backup FEB does not reboot and the interfaces on the FPC connected to the previously active FEB remain online. The backup FEB must obtain the entire forwarding state from the Routing Engine after a switchover, and this update may take a few minutes. If you do not want the interfaces to remain online during the switchover for the other FEB, configure a primary FEB for the redundancy group.

Failover to a backup FEB occurs automatically if a FEB in a redundancy group fails. You can disable automatic failover for any redundancy group by including the **no-auto-failover** statement at the **[edit chassis redundancy feb redundancy-group *group-name*]** hierarchy level.

You can also initiate a manual switchover by issuing the **request chassis redundancy feb slot *slot-number* switch-to-backup** command, where ***slot-number*** is the number of the active FEB. For more information, see the Junos OS Operational Mode Commands.

The following conditions result in failover as long as the backup FEB in a redundancy group is available:

- The FEB is absent.
- The FEB experienced a hard error while coming online.
- A software failure on the FEB resulted in a crash.
- Ethernet connectivity from a FEB to a Routing Engine failed.
- A hard error on the FEB, such as a power failure, occurred.
- The FEB was disabled when the offline button for the FEB was pressed.
- The software watchdog timer on the FEB expired.
- Errors occurred on the links between all the active fabric planes and the FEB. This situation results in failover to the backup FEB if it has at least one valid fabric link.
- Errors occurred on the link between the FEB and all of the FPCs connected to it.

After a switchover occurs, a backup FEB is no longer available for the redundancy group. You can revert from the backup FEB to the previously active FEB by issuing the operational mode command **request chassis redundancy feb slot *slot-number* revert-from-backup**, where ***slot-number*** is the number of the previously active FEB. For more information, see the Junos OS Operational Mode Commands.

When you revert from the backup FEB, it becomes available again for a switchover. If the redundancy group does not have a primary FEB, the backup FEB reboots after you revert back to the previously active FEB. If the FEB to which you revert back is not a primary FEB, the backup FEB is rebooted so that it can aligned with the state of the primary FEB.

If you modify the configuration for an existing redundancy group so that a FEB connects to a different FPC, the FEB is rebooted unless the FEB was already connected to one or two Type 1 FPCs and the change only resulted in the FEB being connected either to one additional or one fewer Type 1 FPC. For more information about how to map a connection between an FPC and a FEB, see the Junos OS System Basics Configuration Guide. If you change the primary FEB in a redundancy group, the backup FEB is rebooted. The FEB is also rebooted if you change a backup FEB to a nonbackup FEB or change an active FEB to a backup FEB.

To view the status of configured FEB redundancy groups, issue the **show chassis redundancy feb** operational mode command. For more information, see the Junos OS Operational Mode Commands.

Redundant SSBs on the M20 Router

The System and Switch Board (SSB) on the M20 router performs the following major functions:

- Shared memory management on the FPCs—The Distributed Buffer Manager ASIC on the SSB uniformly allocates incoming data packets throughout shared memory on the FPCs.
- Outgoing data cell transfer to the FPCs—A second Distributed Buffer Manager ASIC on the SSB passes data cells to the FPCs for packet reassembly when the data is ready to be transmitted.
- Route lookups—The Internet Processor ASIC on the SSB performs route lookups using the forwarding table stored in SSRAM. After performing the lookup, the Internet Processor ASIC informs the midplane of the forwarding decision, and the midplane forwards the decision to the appropriate outgoing interface.
- System component monitoring—The SSB monitors other system components for failure and alarm conditions. It collects statistics from all sensors in the system and relays them to the Routing Engine, which sets the appropriate alarm. For example, if a temperature sensor exceeds the first internally defined threshold, the Routing Engine issues a “high temp” alarm. If the sensor exceeds the second threshold, the Routing Engine initiates a system shutdown.
- Exception and control packet transfer—The Internet Processor ASIC passes exception packets to a microprocessor on the SSB, which processes almost all of them. The remaining packets are sent to the Routing Engine for further processing. Any errors that originate in the Packet Forwarding Engine and are detected by the SSB are sent to the Routing Engine using system log messages.
- FPC reset control—The SSB monitors the operation of the FPCs. If it detects errors in an FPC, the SSB attempts to reset the FPC. After three unsuccessful resets, the SSB takes the FPC offline and informs the Routing Engine. Other FPCs are unaffected, and normal system operation continues.

The M20 router holds up to two SSBs. One SSB is configured to act as the master and the other is configured to serve as a backup in case the master fails. You can initiate a manual switchover by issuing the **request chassis ssb master switch** command. For more information, see the Junos OS Operational Mode Commands.

Redundant SFMs on the M40e and M160 Routers

The M40e and M160 routers have redundant Switching and Forwarding Modules (SFMs). The SFMs contain the Internet Processor II ASIC and two Distributed Buffer Manager ASICs. SFMs ensure that all traffic leaving the FPCs is handled properly. SFMs provide route lookup, filtering, and switching.

The M40e router holds up to two SFMs, one that is configured to act as the master and the other configured to serve as a backup in case the master fails. Removing the standby SFM has no effect on router function. If the active SFM fails or is removed from the chassis, forwarding halts until the standby SFM boots and becomes active. It takes approximately 1 minute for the new SFM to become active. Synchronizing router configuration information can take additional time, depending on the complexity of the configuration.

The M160 router holds up to four SFMs. All SFMs are active at the same time. A failure or taking an SFM offline has no effect on router function. Forwarding continues uninterrupted.

You can initiate a manual switchover by issuing the **request chassis sfm master switch** command. For more information, see the Junos OS Operational Mode Commands.

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Understanding Routing Engine Redundancy on Juniper Networks Routers on page 11
- Configuring CFEB Redundancy on the M10i Router on page 29
- Configuring FEB Redundancy on the M120 Router on page 30
- Configuring SFM Redundancy on M40e and M160 Routers on page 32
- Configuring SSB Redundancy on the M20 Router on page 32

- show chassis redundancy feb

CHAPTER 3

Routing Engine and Switching Control Board Redundancy Configuration Guidelines

This chapter includes the following topics:

Chassis Redundancy Hierarchy

The following redundancy statements are available at the **[edit chassis]** hierarchy level:

```
redundancy {
  cfeb slot (always | preferred);
  failover {
    on-disk-failure;
    on-loss-of-keepalives;
  }
  feb {
    redundancy-group group-name {
      description description;
      feb slot-number (backup | primary);
      no-auto-failover;
    }
  }
  graceful-switchover;
  keepalive-time seconds;
```

```
    routing-engine slot-number (master | backup | disabled);
    sfm slot-number (always | preferred);
    ssb slot-number (always | preferred);
}
```

Related Documentation

- Configuring Routing Engine Redundancy on page 25
- Configuring CFEB Redundancy on the M10i Router on page 29
- Configuring FEB Redundancy on the M120 Router on page 30
- Configuring SFM Redundancy on M40e and M160 Routers on page 32
- Configuring SSB Redundancy on the M20 Router on page 32

Initial Routing Engine Configuration Example

You can use configuration groups to ensure that the correct IP addresses are used for each Routing Engine and to maintain a single configuration file for both Routing Engines.

The following example defines configuration groups **re0** and **re1** with separate IP addresses. These well-known configuration group names take effect only on the appropriate Routing Engine.

```
groups {
  re0 {
    system {
      host-name my-re0;
    }
    interfaces {
      fxp0 {
        description "10/100 Management interface";
        unit 0 {
          family inet {
            address 10.255.2.40/24;
          }
        }
      }
    }
  }
  re1 {
    system {
      host-name my-re1;
    }
    interfaces {
      fxp0 {
        description "10/100 Management interface";
        unit 0 {
          family inet {
            address 10.255.2.41/24;
          }
        }
      }
    }
  }
}
```


You can assign an additional IP address to the management Ethernet interface (**fxp0** in this example) on both Routing Engines. The assigned address uses the **master-only** keyword and is identical for both Routing Engines, ensuring that the IP address for the master Routing Engine can be accessed at any time. The address is active only on the master Routing Engine's management Ethernet interface. During a Routing Engine switchover, the address moves over to the new master Routing Engine.

For example, on **re0**, the configuration is:

```
[edit groups re0 interfaces fxp0]
unit 0 {
  family inet {
    address 10.17.40.131/25 {
      master-only;
    }
    address 10.17.40.132/25;
  }
}
```

On **re1**, the configuration is:

```
[edit groups re1 interfaces fxp0]
unit 0 {
  family inet {
    address 10.17.40.131/25 {
      master-only;
    }
    address 10.17.40.133/25;
  }
}
```

For more information about the initial configuration of dual Routing Engines, see the Installation and Upgrade Guide. For more information about assigning an additional IP address to the management Ethernet interface with the **master-only** keyword on both Routing Engines, see the CLI User Guide.

Related Documentation

- Understanding Routing Engine Redundancy on Juniper Networks Routers on page 11
- Switching Control Board Redundancy on page 15

Copying a Configuration File from One Routing Engine to the Other

You can use either the console port or the management Ethernet port to establish connectivity between the two Routing Engines. You can then copy or use FTP to transfer the configuration from the master to the backup, and load the file and commit it in the normal way.

To connect to the other Routing Engine using the management Ethernet port, issue the following command:

```
user@host> request routing-engine login (other-routing-engine | re0 | re1)
```


On a TX Matrix router, to make connections to the other Routing Engine using the management Ethernet port, issue the following command:

user@host> **request routing-engine login (backup | lcc *number* | master | other-routing-engine | re0 | re1)**

For more information about the **request routing-engine login** command, see the Junos OS Operational Mode Commands.

To copy a configuration file from one Routing Engine to the other, issue the **file copy** command:

user@host> **file copy *source destination***

In this case, ***source*** is the name of the configuration file. These files are stored in the directory **/config**. The active configuration is **/config/juniper.conf**, and older configurations are in **/config/juniper.conf {1...9}**. The ***destination*** is a file on the other Routing Engine.

The following example copies a configuration file from Routing Engine 0 to Routing Engine 1:

user@host> **file copy /config/juniper.conf re1:/var/tmp/copied-juniper.conf**

The following example copies a configuration file from Routing Engine 0 to Routing Engine 1 on a TX Matrix router:

user@host> **file copy /config/juniper.conf scc-re1:/var/tmp/copied-juniper.conf**

To load the configuration file, enter the **load replace** command at the **[edit]** hierarchy level:

user@host> **load replace /var/tmp/copied-juniper.conf**

CAUTION: **Make sure you change any IP addresses specified in the management Ethernet interface configuration on Routing Engine 0 to addresses appropriate for Routing Engine 1.**

Related Documentation

- Understanding Routing Engine Redundancy on Juniper Networks Routers on page 11
- Switching Control Board Redundancy on page 15
- Loading a Software Package from the Other Routing Engine on page 24

Loading a Software Package from the Other Routing Engine

You can load a package from the other Routing Engine onto the local Routing Engine using the existing **request system software add *package-name*** command:

user@host> **request system software add re(0|1):/*filename***

In the **re** portion of the URL, specify the number of the other Routing Engine. In the ***filename*** portion of the URL, specify the path to the package. Packages are typically in the directory **/var/sw/pkg**.

Related Documentation

Configuring Routing Engine Redundancy

The following sections describe how to configure Routing Engine redundancy:

NOTE: To complete the tasks in the following sections, **re0** and **re1** configuration groups must be defined. For more information about configuration groups, see the CLI User Guide.

Modifying the Default Routing Engine Mastership

For routers with two Routing Engines, you can configure which Routing Engine is the master and which is the backup. By default, the Routing Engine in slot 0 is the master (**re0**) and the one in slot 1 is the backup (**re1**).

NOTE: In systems with two Routing Engines, both Routing Engines cannot be configured to be master at the same time. This configuration causes the commit check to fail.

To modify the default configuration, include the **routing-engine** statement at the **[edit chassis redundancy]** hierarchy level:

```
[edit chassis redundancy]
routing-engine slot-number (master | backup | disabled);
```

slot-number can be 0 or 1. To configure the Routing Engine to be the master, specify the **master** option. To configure it to be the backup, specify the **backup** option. To disable a Routing Engine, specify the **disabled** option.

NOTE: To switch between the master and the backup Routing Engines, you must modify the configuration and then activate it by issuing the **commit synchronize** command.

Configuring Automatic Failover to the Backup Routing Engine

The following sections describe how to configure automatic failover to the backup Routing Engine when certain failures occur on the master Routing Engine.

Without Interruption to Packet Forwarding

For routers with two Routing Engines, you can configure graceful Routing Engine switchover (GRES). When graceful switchover is configured, socket reconnection occurs seamlessly without interruption to packet forwarding. For information about how to configure graceful Routing Engine switchover, see "Configuring Graceful Routing Engine Switchover" on page 59.

On Detection of a Hard Disk Error on the Master Routing Engine

After you configure a backup Routing Engine, you can direct it to take mastership automatically if it detects a hard disk error from the master Routing Engine. To enable this feature, include the **on-disk-failure** statement at the **[edit chassis redundancy failover]** hierarchy level.

```
[edit chassis redundancy failover]
on-disk-failure;
```

On Detection of a Loss of Keepalive Signal from the Master Routing Engine

After you configure a backup Routing Engine, you can direct it to take mastership automatically if it detects a loss of keepalive signal from the master Routing Engine.

To enable failover on receiving a loss of keepalive signal, include the **on-loss-of-keepalives** statement at the **[edit chassis redundancy failover]** hierarchy level:

```
[edit chassis redundancy failover]
on-loss-of-keepalives;
```

When graceful Routing Engine switchover is not configured, by default, failover occurs after 300 seconds (5 minutes). You can configure a shorter or longer time interval.

NOTE: **The keepalive time period is reset to 360 seconds when the master Routing Engine has been manually rebooted or halted.**

To change the keepalive time period, include the **keepalive-time** statement at the **[edit chassis redundancy]** hierarchy level:

```
[edit chassis redundancy]
keepalive-time seconds;
```

The range for **keepalive-time** is 2 through 10,000 seconds.

The following example describes the sequence of events if you configure the backup Routing Engine to detect a loss of keepalive signal in the master Routing Engine:

1. Manually configure a **keepalive-time** of 25 seconds.
2. After the Packet Forwarding Engine connection to the primary Routing Engine is lost and the keepalive timer expires, packet forwarding is interrupted.
3. After 25 seconds of keepalive loss, a message is logged, and the backup Routing Engine attempts to take mastership. An alarm is generated when the backup Routing Engine becomes active, and the display is updated with the current status of the Routing Engine.
4. After the backup Routing Engine takes mastership, it continues to function as master.

NOTE: When graceful Routing Engine switchover is configured, the keepalive signal is automatically enabled and the failover time is set to 2 seconds (4 seconds on M20 routers). You cannot manually reset the keepalive time.

NOTE: When you halt or reboot the master Routing Engine, Junos OS resets the keepalive time to 360 seconds, and the backup Routing Engine does not take over mastership until the 360-second keepalive time period expires.

A former master Routing Engine becomes a backup Routing Engine if it returns to service after a failover to the backup Routing Engine. To restore master status to the former master Routing Engine, you can use the **request chassis routing-engine master switch** operational mode command.

If at any time one of the Routing Engines is not present, the remaining Routing Engine becomes master automatically, regardless of how redundancy is configured.

When a Software Process Fails

To configure automatic switchover to the backup Routing Engine if a software process fails, include the **failover other-routing-engine** statement at the **[edit system processes *process-name*]** hierarchy level:

```
[edit system processes process-name]
failover other-routing-engine;
```

process-name is one of the valid process names. If this statement is configured for a process, and that process fails four times within 30 seconds, the router reboots from the other Routing Engine. Another statement available at the **[edit system processes]** hierarchy level is **failover alternate-media**. For information about the alternate media option, see the Junos OS System Basics Configuration Guide.

Manually Switching Routing Engine Mastership

To manually switch Routing Engine mastership, use one of the following commands:

- On the backup Routing Engine, request that the backup Routing Engine take mastership by issuing the **request chassis routing-engine master acquire** command.
- On the master Routing Engine, request that the backup Routing Engine take mastership by using the **request chassis routing-engine master release** command.
- On either Routing Engine, switch mastership by issuing the **request chassis routing-engine master switch** command.

Verifying Routing Engine Redundancy Status

A separate log file is provided for redundancy logging at **/var/log/mastership**. To view the log, use the **file show /var/log/mastership** command. Table 3 on page 28 lists the mastership log event codes and descriptions.

Table 3: Routing Engine Mastership Log

Event Code	Description
E_NULL = 0	The event is a null event.
E_CFG_M	The Routing Engine is configured as master.
E_CFG_B	The Routing Engine is configured as backup.
E_CFG_D	The Routing Engine is configured as disabled.
E_MAXTRY	The maximum number of tries to acquire or release mastership was exceeded.
E_REQ_C	A claim mastership request was sent.
E_ACK_C	A claim mastership acknowledgement was received.
E_NAK_C	A claim mastership request was not acknowledged.
E_REQ_Y	Confirmation of mastership is requested.
E_ACK_Y	Mastership is acknowledged.
E_NAK_Y	Mastership is not acknowledged.
E_REQ_G	A release mastership request was sent by a Routing Engine.
E_ACK_G	The Routing Engine acknowledged release of mastership.
E_CMD_A	The command **request chassis routing-engine master acquire** was issued from the backup Routing Engine.

Table 3: Routing Engine Mastership Log *(continued)*

Event Code	Description
E_CMD_F	The command **request chassis routing-engine master acquire force** was issued from the backup Routing Engine.
E_CMD_R	The command **request chassis routing-engine master release** was issued from the master Routing Engine.
E_CMD_S	The command request **chassis routing-engine master switch** was issued from a Routing Engine.
E_NO_ORE	No other Routing Engine is detected.
E_TMOUT	A request timed out.
E_NO_IPC	Routing Engine connection was lost.
E_ORE_M	Other Routing Engine state was changed to master.
E_ORE_B	Other Routing Engine state was changed to backup.
E_ORE_D	Other Routing Engine state was changed to disabled.

Related Documentation

- Understanding Routing Engine Redundancy on Juniper Networks Routers on page 11
- Switching Control Board Redundancy on page 15
- Chassis Redundancy Hierarchy on page 21

Configuring CFEB Redundancy on the M10i Router

The Compact Forwarding Engine Board (CFEB) on the M10i router provides route lookup, filtering, and switching on incoming data packets, and then directs outbound packets to the appropriate interface for transmission to the network. The CFEB communicates with the Routing Engine using a dedicated 100-Mbps Fast Ethernet link that transfers routing table data from the Routing Engine to the forwarding table in the integrated ASIC. The link is also used to transfer from the CFEB to the Routing Engine routing link-state updates and other packets destined for the router that have been received through the router interfaces.

To configure a CFEB redundancy group, include the following statements at the **[edit chassis redundancy]** hierarchy level:

```
[edit chassis redundancy]
cfeb slot-number (always | preferred);
```

slot-number can be 0 or 1.

always defines the CFEB as the sole device.

preferred defines a preferred CFEB.

To manually switch CFEB mastership, issue the **request chassis cfeb master switch** command. To view CFEB status, issue the **show chassis cfeb** command.

Related Documentation

- Switching Control Board Redundancy on page 15
- Chassis Redundancy Hierarchy on page 21

Configuring FEB Redundancy on the M120 Router

To configure a FEB redundancy group for the M120 router, include the following statements at the **[edit chassis redundancy feb]** hierarchy level:

```
[edit chassis redundancy feb]
redundancy-group group-name {
  description description;
  feb slot-number (backup | primary);
  no-auto-failover;
}
```

group-name is the unique name for the redundancy group. The maximum length is 39 alphanumeric characters.

slot-number is the slot number of each FEB you want to include in the redundancy group. The range is from 0 through 5. You must specify exactly one FEB as a backup FEB per redundancy group. Include the **backup** keyword when configuring the backup FEB and make sure that the FEB is not connected to an FPC.

Include the **primary** keyword to optionally specify one primary FEB per redundancy group. When the **primary** keyword is specified for a particular FEB, that FEB is configured for 1:1 redundancy. With 1:1 redundancy, the backup FEB contains the same forwarding state as the primary FEB. When no FEB in the redundancy group is configured as a primary FEB, the redundancy group is configured for *n*:1 redundancy. In this case, the backup FEB has no forwarding state. When a FEB fails, the forwarding state must be downloaded from the Routing Engine to the backup FEB before forwarding continues.

A combination of 1:1 and *n*:1 redundancy is possible when more than two FEBs are present in a group. The backup FEB contains the same forwarding state as the primary FEB, so that when the primary FEB fails, 1:1 failover is in effect. When a nonprimary FEB fails, the backup FEB must be rebooted so that the forwarding state from the nonprimary FEB is installed on the backup FEB before it can continue forwarding.

You can optionally include the **description** statement to describe a redundancy group.

Automatic failover is enabled by default. To disable automatic failover, include the **no-auto-failover** statement. If you disable automatic failover, you can perform only a manual switchover using the operational command **request chassis redundancy feb slot *slot-number* switch-to-backup**.

To view FEB status, issue the **show chassis feb** command. For more information, see the Junos OS Operational Mode Commands.

Related Documentation

- Switching Control Board Redundancy on page 15
- Chassis Redundancy Hierarchy on page 21
- Example: Configuring FEB Redundancy on page 31

Example: Configuring FEB Redundancy

In the following configuration, two FEB redundancy groups are created:

- A FEB redundancy group named **group0** with the following properties:
  - Contains three FEBs (0 through 2).
  - Has a primary FEB (2).
  - Has a unique backup FEB (0).
  - Automatic failover is disabled.

    When an active FEB in **group0** fails, automatic failover to the backup FEB does not occur. For **group0**, you can only perform a manual switchover.
- A FEB redundancy group named **group1** with the following properties:
  - Two FEBs (3 and 5). There is no primary FEB.
  - A unique backup FEB (5).
  - Automatic failover is enabled by default.

    When **feb 3** in **group1** fails, an automatic failover occurs.

Because you must explicitly configure an FPC *not* to connect to the backup FEB, connectivity is set to none between **fpc 0** and **feb 0** and between **fpc 5** and **feb 5**.

NOTE: For information about the **fpc-feb-connectivity** statement, see the Junos OS System Basics Configuration Guide.

FPC to primary FEB connectivity is not explicitly configured, so by default, the software automatically assigns connectivity based on the numerical order of the FPCs.

```
[edit]
chassis {
  fpc-feb-connectivity {
    fpc 0 feb none;
    fpc 5 feb none;
  }
  redundancy feb {
    redundancy-group group0 {
      description “Interfaces to Customer X”;
      feb 2 primary;
      feb 1;
      feb 0 backup;
      no-auto-failover;
```

```
    }
    redundancy-group group1 {
      feb 3;
      feb 5 backup;
    }
  }
```

Related Documentation

- Switching Control Board Redundancy on page 15
- Configuring FEB Redundancy on the M120 Router on page 30

Configuring SFM Redundancy on M40e and M160 Routers

By default, the Switching and Forwarding Module (SFM) in slot 0 is the master and the SFM in slot 1 is the backup. To modify the default configuration, include the **sfm** statement at the **[edit chassis redundancy]** hierarchy level:

```
[edit chassis redundancy]
sfm slot-number (always | preferred);
```

On the M40e router, ***slot-number*** is 0 or 1. On the M160 router, ***slot-number*** is 0 through 3.

always defines the SFM as the sole device.

preferred defines a preferred SFM.

To manually switch mastership between SFMs, issue the **request chassis sfm master switch** command. To view SFM status, issue the **show chassis sfm** command. For more information, see the Junos OS Operational Mode Commands.

Related Documentation

- Switching Control Board Redundancy on page 15
- Chassis Redundancy Hierarchy on page 21

Configuring SSB Redundancy on the M20 Router

For M20 routers with two System and Switch Boards (SSBs), you can configure which SSB is the master and which is the backup. By default, the SSB in slot 0 is the master and the SSB in slot 1 is the backup. To modify the default configuration, include the **ssb** statement at the **[edit chassis redundancy]** hierarchy level:

```
[edit chassis redundancy]
ssb slot-number (always | preferred);
```

slot-number is 0 or 1.

always defines the SSB as the sole device.

preferred defines a preferred SSB.

To manually switch mastership between SSBs, issue the **request chassis ssb master switch** command.

To display SSB status information, issue the **show chassis ssb** command. The command output displays the number of times the mastership has changed, the SSB slot number, and the current state of the SSB: master, backup, or empty. For more information, see the Junos OS Operational Mode Commands.

Related Documentation

- Switching Control Board Redundancy on page 15
- Chassis Redundancy Hierarchy on page 21

CHAPTER 4

Summary of Routing Engine and Switching Control Board Redundancy Statements

This chapter provides a reference for each of the Routing Engine and switching control board redundancy configuration statements. The statements are organized alphabetically.

cfeb

Syntax cfeb *slot-number* (always | preferred);

Hierarchy Level [edit chassis redundancy]

Release Information Statement introduced before Junos OS Release 7.4.

Description On M10i routers only, configure which Compact Forwarding Engine Board (CFEB) is the master and which is the backup.

Default By default, the CFEB in slot 0 is the master and the CFEB in slot 1 is the backup.

Options ***slot-number***—Specify which slot is the master and which is the backup.

always—Define this CFEB as the sole device.

preferred—Define this CFEB as the preferred device of at least two.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring CFEB Redundancy on the M10i Router on page 29

description (Chassis Redundancy)

Syntax description *description*;

Hierarchy Level [edit chassis redundancy feb redundancy-group *group-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Provide a description of the FEB redundancy group.

Options *description*—Provide a description for the FEB redundancy group.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring FEB Redundancy on the M120 Router on page 30

failover (Chassis)

Syntax

```
failover {
  on-disk-failure;
  on-loss-of-keepalives;
}
```

Hierarchy Level [edit chassis redundancy]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify conditions on the master Routing Engine that cause the backup router to take mastership.

The remaining statement are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- On Detection of a Hard Disk Error on the Master Routing Engine on page 26

failover (System Process)

Syntax failover (alternate-media | other-routing-engine);

Hierarchy Level [edit system processes *process-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the router to reboot if the software process fails four times within 30 seconds, and specify the software to use during the reboot.

Options ***process-name***—Junos OS process name. Some of the processes that support the **failover** statement are **bootp**, **chassis-control**, **craft-control**, **ethernet-connectivity-fault-management**, **init**, **interface-control**, **neighbor-liveness**, **pfe**, **redundancy-interface-process**, **routing**, and **vrrp**.

alternate-media—Use the Junos OS image on alternate media during the reboot.

other-routing-engine—On routers with dual Routing Engines, use the Junos OS image on the other Routing Engine during the reboot. That Routing Engine assumes mastership; in the usual configuration, the other Routing Engine is the designated backup Routing Engine.

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

Related Documentation

- When a Software Process Fails on page 27
- processes

feb

feb (Creating a Redundancy Group)

Syntax

```
feb {
  redundancy-group group-name {
    description description;
    feb slot-number (backup | primary);
    no-auto-failover;
  }
}
```

Hierarchy Level [edit chassis redundancy]

Release Information Statement introduced in Junos OS Release 8.2.

Description On M120 routers only, configure a Forwarding Engine Board (FEB) redundancy group.

Options The remaining statements are described separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring FEB Redundancy on the M120 Router on page 30

feb (Assigning a FEB to a Redundancy Group)

Syntax
feb *slot-number* (backup | primary);

Hierarchy Level
[edit chassis redundancy feb redundancy-group *group-name*]

Release Information
Statement introduced in Junos OS Release 8.2.

Description
On M120 routers only, configure a Forwarding Engine Board (FEB) as part of a FEB redundancy group.

Options
slot-number—Slot number of the FEB. The range of values is from **0** to **5**.

backup—(Optional) For each redundancy group, you must configure exactly one backup FEB.

primary—(Optional) For each redundancy group, you can optionally configure one primary FEB.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring FEB Redundancy on the M120 Router on page 30

keepalive-time

Syntax keepalive-time *seconds*;

Hierarchy Level [edit chassis redundancy]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the time period that must elapse before the backup router takes mastership when it detects loss of the keepalive signal.

Default The **on-loss-of-keepalives** statement at the **[edit chassis redundancy failover]** hierarchy level must be included for failover to occur.

When the **on-loss-of-keepalives** statement is included and graceful Routing Engine switchover *is not* configured, failover occurs after 300 seconds (5 minutes).

When the **on-loss-of-keepalives** statement is included and graceful Routing Engine switchover *is* configured, the keepalive signal is automatically enabled and the failover time is set to 2 seconds (4 seconds on M20 routers). You cannot manually reset the keepalive time.

Options *seconds*—Time before the backup router takes mastership when it detects loss of the keepalive signal. The range of values is 2 through 10,000.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- On Detection of a Loss of Keepalive Signal from the Master Routing Engine on page 26
- failover (Chassis) on page 36
- on-loss-of-keepalives on page 42

no-auto-failover

Syntax no-auto-failover;

Hierarchy Level [edit chassis redundancy feb redundancy-group *group-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Disable automatic failover to a backup FEB when an active FEB in a redundancy group fails.

Default Automatic failover is enabled by default.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring FEB Redundancy on the M120 Router on page 30

on-disk-failure (Chassis Redundancy Failover)

Syntax on-disk-failure;

Hierarchy Level [edit chassis redundancy failover]

Release Information Statement introduced before Junos OS Release 7.4.

Description Instruct the backup router to take mastership if it detects hard disk errors on the master Routing Engine.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- On Detection of a Hard Disk Error on the Master Routing Engine on page 26

on-loss-of-keepalives

Syntax on-loss-of-keepalives;

Hierarchy Level [edit chassis redundancy failover]

Release Information Statement introduced before Junos OS Release 7.4.

Description Instruct the backup router to take mastership if it detects a loss of keepalive signal from the master Routing Engine.

Default The **on-loss-of-keepalives** statement must be included at the **[edit chassis redundancy failover]** hierarchy level for failover to occur.

When the **on-loss-of-keepalives** statement is included but graceful Routing Engine switchover *is not* configured, failover occurs after 300 seconds (5 minutes).

When the **on-loss-of-keepalives** statement is included and graceful Routing Engine switchover *is* configured, the keepalive signal is automatically enabled and the failover time is set to 2 seconds (4 seconds on M20 routers) . The keepalive time is not configurable.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- On Detection of a Loss of Keepalive Signal from the Master Routing Engine on page 26
- **keepalive-time on page 40**

redundancy

Syntax

```
redundancy {
  cfeb slot (always | preferred);
  failover {
    on-disk-failure;
    on-loss-of-keepalives;
  }
  feb {
    redundancy-group group-name {
      description description;
      feb slot-number (backup | primary);
      no-auto-failover;
    }
  }
  graceful-switchover;
  keepalive-time seconds;
  routing-engine slot-number (backup | disabled | master);
  sfm slot-number (always | preferred);
  ssb slot-number (always | preferred);
}
```

Hierarchy Level [edit chassis]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure redundancy options.

Options The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Chassis Redundancy Hierarchy on page 21
- Configuring Routing Engine Redundancy on page 25
- Configuring CFEB Redundancy on the M10i Router on page 29
- Configuring FEB Redundancy on the M120 Router on page 30
- Configuring SFM Redundancy on M40e and M160 Routers on page 32
- Configuring SSB Redundancy on the M20 Router on page 32

redundancy-group

Syntax

```
redundancy-group group-name {
  description description;
  feb slot-number (backup | primary);
  no-auto-failover;
}
```

Hierarchy Level [edit chassis redundancy feb]

Release Information Statement introduced in Junos OS Release 8.2.

Description On M120 routers only, configure a Forwarding Engine Board (FEB) redundancy group.

Options ***group-name*** is the unique name for the redundancy group. The maximum length is 39 alphanumeric characters.

Other statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring FEB Redundancy on the M120 Router on page 30

routing-engine (Chassis Redundancy)

Syntax routing-engine *slot-number* (backup | disabled | master);

Hierarchy Level [edit chassis redundancy]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure Routing Engine redundancy.

Default By default, the Routing Engine in slot 0 is the master Routing Engine and the Routing Engine in slot 1 is the backup Routing Engine.

Options ***slot-number***—Specify the slot number (0 or 1).

Set the function of the Routing Engine for the specified slot:

- **master**—Routing Engine in the specified slot is the master.
- **backup**—Routing Engine in the specified slot is the backup.
- **disabled**—Routing Engine in the specified slot is disabled.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Routing Engine Redundancy on page 25

sfm (Chassis Redundancy)

Syntax sfm *slot-number* (always | preferred);

Hierarchy Level [edit chassis redundancy]

Release Information Statement introduced before Junos OS Release 7.4.

Description On M40e and M160 routers, configure which Switching and Forwarding Module (SFM) is the master and which is the backup.

Default By default, the SFM in slot 0 is the master and the SFM in slot 1 is the backup.

Options ***slot-number***—Specify which slot is the master and which is the backup. On the M40e router, ***slot-number*** can be 0 or 1. On the M160 router, ***slot-number*** can be 0 through 3.

always—Define this SFM as the sole device.

preferred—Define this SFM as the preferred device of at least two.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring SFM Redundancy on M40e and M160 Routers on page 32

ssb

Syntax ssb *slot-number* (always | preferred);

Hierarchy Level [edit chassis redundancy]

Release Information Statement introduced before Junos OS Release 7.4.

Description On M20 routers, configure which System and Switch Board (SSB) is the master and which is the backup.

Default By default, the SSB in slot 0 is the master and the SSB in slot 1 is the backup.

Options ***slot-number***—Specify which slot is the master and which is the backup.

always—Define this SSB as the sole device.

preferred—Define this SSB as the preferred device of at least two.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring SSB Redundancy on the M20 Router on page 32

PART 3

Graceful Routing Engine Switchover

CHAPTER 5

Graceful Routing Engine Switchover Overview

This chapter contains the following sections:

Understanding Graceful Routing Engine Switchover in the Junos OS

This topic contains the following sections:

Graceful Routing Engine Switchover Concepts

Graceful Routing Engine switchover (GRES) feature in Junos OS enables a routing platform with redundant Routing Engines to continue forwarding packets, even if one Routing Engine fails. Graceful Routing Engine switchover preserves interface and kernel information. Traffic is not interrupted. However, graceful Routing Engine switchover does not preserve the control plane. Neighboring routers detect that the router has experienced a restart and react to the event in a manner prescribed by individual routing protocol specifications. To preserve routing during a switchover, graceful Routing Engine switchover must be combined with either graceful restart protocol extensions or nonstop active routing. Any updates to the master Routing Engine are replicated to the backup Routing Engine as soon as they occur. If the kernel on the master Routing Engine stops operating, the master Routing Engine experiences a hardware failure, or the administrator initiates a manual switchover, mastership switches to the backup Routing Engine.

NOTE: To quickly restore or to preserve routing protocol state information during a switchover, graceful Routing Engine switchover must be combined with either graceful restart or nonstop active routing (NSR), respectively. For more information about graceful restart, see “Graceful Restart Concepts” on page 105. For more information about nonstop active routing, see “Nonstop Active Routing Concepts” on page 77.

If the backup Routing Engine does not receive a keepalive from the master Routing Engine after 2 seconds (4 seconds on M20 routers), it determines that the master Routing Engine has failed and takes mastership. The Packet Forwarding Engine seamlessly disconnects from the old master Routing Engine and reconnects to the new master Routing Engine. The Packet Forwarding Engine does not reboot, and traffic is not interrupted. The new master Routing Engine and the Packet Forwarding Engine then become synchronized. If the new master Routing Engine detects that the Packet Forwarding Engine state is not up to date, it resends state update messages.

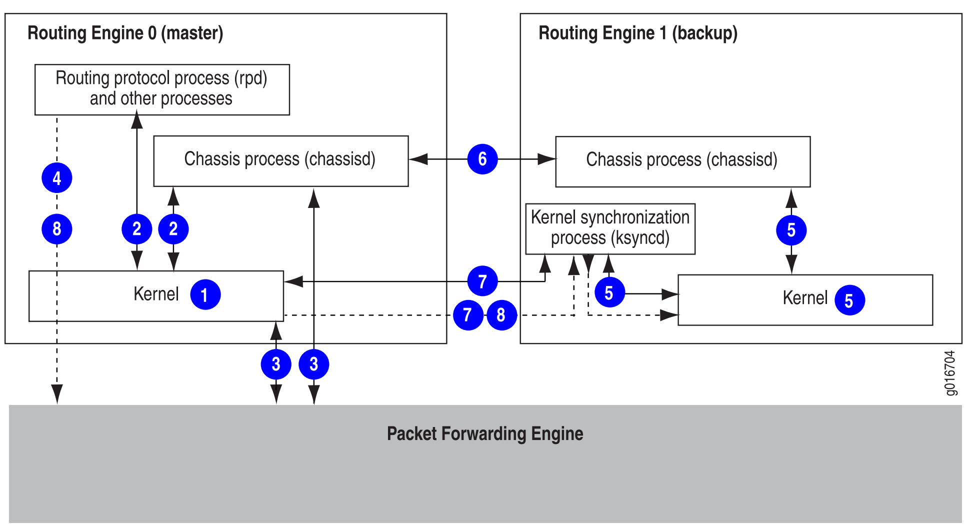
NOTE: Successive Routing Engine switchover events must be a minimum of 240 seconds (4 minutes) apart after both Routing Engines have come up.

If the router displays a warning message similar to “Standby Routing Engine is not ready for graceful switchover. Packet Forwarding Engines that are not ready for graceful switchover might be reset,” do not attempt switchover. If you choose to proceed with switchover, only the Packet Forwarding Engines that were not ready for graceful switchover are reset. None of the FPCs should spontaneously restart. We recommend that you wait until the warning no longer appears and then proceed with the switchover.

NOTE: We do not recommend performing a commit operation on the backup Routing Engine when graceful Routing Engine switchover is enabled on the router.

Figure 1 on page 52 shows the system architecture of graceful Routing Engine switchover and the process a routing platform follows to prepare for a switchover.

Figure 1: Preparing for a Graceful Routing Engine Switchover

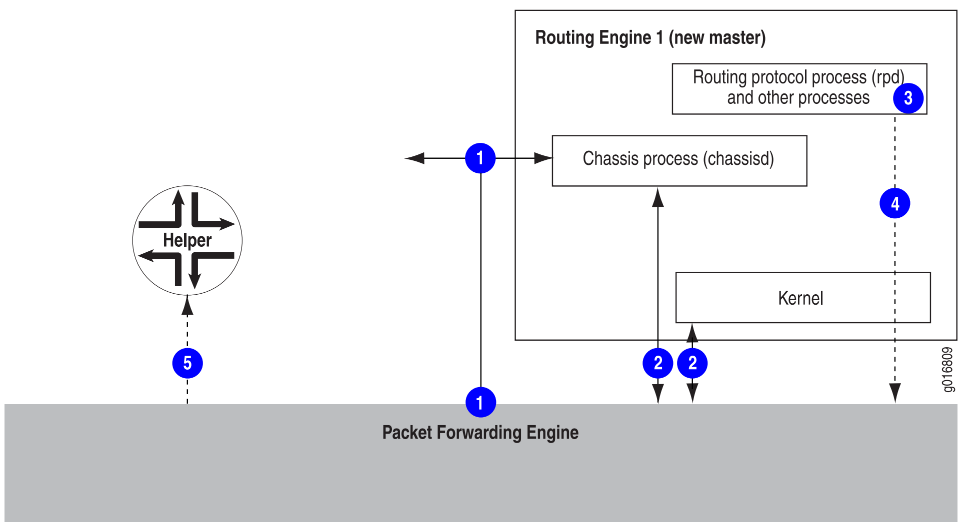


The switchover preparation process for graceful Routing Engine switchover follows these steps:

1. The master Routing Engine starts.
2. The routing platform processes (such as the chassis process [chassisd]) start.
3. The Packet Forwarding Engine starts and connects to the master Routing Engine.
4. All state information is updated in the system.
5. The backup Routing Engine starts.
6. The system determines whether graceful Routing Engine switchover has been enabled.
7. The kernel synchronization process (ksyncd) synchronizes the backup Routing Engine with the master Routing Engine.
8. After ksyncd completes the synchronization, all state information and the forwarding table are updated.

Figure 2 on page 53 shows the effects of a switchover on the routing platform.

Figure 2: Graceful Routing Engine Switchover Process



When a switchover occurs, the switchover process follows these steps:

1. When keepalives from the master Routing Engine are lost, the system switches over gracefully to the backup Routing Engine.
2. The Packet Forwarding Engine connects to the backup Routing Engine, which becomes the new master.
3. Routing platform processes that are not part of graceful Routing Engine switchover (such as the routing protocol process [rpd]) restart.
4. State information learned from the point of the switchover is updated in the system.
5. If configured, graceful restart protocol extensions collect and restore routing information from neighboring peer *helper* routers.

NOTE: On T Series and M320 routers, the Switch Interface Boards (SIBs) are taken offline and restarted one by one during a graceful Routing Engine switchover. This is done to provide the SPMB that manages the SIB enough time to populate state information for its associated SIB. However, on a fully-populated chassis where all FPCs are sending traffic at full line rate, there might be momentary packet loss during the switchover.

Effects of a Routing Engine Switchover

Table 4 on page 54 describes the effects of a Routing Engine switchover when no high availability features are enabled and when graceful Routing Engine switchover, graceful restart, and nonstop active routing features are enabled.

Table 4: Effects of a Routing Engine Switchover

Feature	Benefits	Considerations
Dual Routing Engines only (no features enabled)	When the switchover to the new master Routing Engine is complete, routing convergence takes place and traffic is resumed.	All physical interfaces are taken offline, Packet Forwarding Engines restart, the standby Routing Engine restarts the routing protocol process (rpd), and all hardware and interfaces are discovered by the new master Routing Engine. The switchover takes several minutes and all of the router's adjacencies are aware of the physical (interface alarms) and routing (topology) change.
Graceful Routing Engine switchover enabled	During the switchover, interface and kernel information is preserved. The switchover is faster because the Packet Forwarding Engines are not restarted.	The new master Routing Engine restarts the routing protocol process (rpd). All hardware and interfaces are acquired by a process that is similar to a warm restart. All adjacencies are aware of the router's change in state.
Graceful Routing Engine switchover and nonstop active routing enabled	Traffic is not interrupted during the switchover. Interface, kernel, and routing protocol information is preserved.	Unsupported protocols must be refreshed using the normal recovery mechanisms inherent in each protocol.
Graceful Routing Engine switchover and graceful restart enabled	Traffic is not interrupted during the switchover. Interface and kernel information is preserved. Graceful restart protocol extensions quickly collect and restore routing information from the neighboring routers.	Neighbors are required to support graceful restart and a wait interval is required. The routing protocol process (rpd) restarts. For certain protocols, a significant change in the network can cause graceful restart to stop.

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Graceful Routing Engine Switchover System Requirements on page 55
- Configuring Graceful Routing Engine Switchover on page 59
- Requirements for Routers with a Backup Router Configuration on page 60

Graceful Routing Engine Switchover System Requirements

Graceful Routing Engine switchover is supported on all routing platforms that contain dual Routing Engines. All Routing Engines configured for graceful Routing Engine switchover must run the same Junos OS Release. Hardware and software support for graceful Routing Engine switchover is described in the following sections:

Graceful Routing Engine Switchover Platform Support

To enable graceful Routing Engine switchover, your system must meet these minimum requirements:

- M20 and M40e routers—Junos OS Release 5.7 or later
- M10i router—Junos OS Release 6.1 or later
- M320 router—Junos OS Release 6.2 or later
- T320 router, T640 router, and TX Matrix router—Junos OS Release 7.0 or later
- M120 router—Junos OS Release 8.2 or later
- MX960 router—Junos OS Release 8.3 or later
- MX480 router—Junos OS Release 8.4 or later (8.4R2 recommended)
- MX240 router—Junos OS Release 9.0 or later
- T1600 router—Junos OS Release 8.5 or later
- T4000 router—Junos OS Release 12.1R2 or later
- TX Matrix Plus router—Junos OS Release 9.6 or later

For more information about support for graceful Routing Engine switchover, see the sections that follow.

Graceful Routing Engine Switchover Feature Support

Graceful Routing Engine switchover supports most Junos OS features in Release 5.7 and later. Particular Junos OS features require specific versions of Junos OS. See Table 5 on page 55.

Table 5: Graceful Routing Engine Switchover Feature Support

Application	Junos OS Release
Aggregated Ethernet interfaces with Link Aggregation Control Protocol (LACP) and aggregated SONET interfaces	6.2

Table 5: Graceful Routing Engine Switchover Feature Support *(continued)*

Application	Junos OS Release
Asynchronous Transfer Mode (ATM) virtual circuits (VCs)	6.2
Logical systems NOTE: In Junos OS Release 9.3 and later, the logical router feature is renamed to logical system.	6.3
Multicast	6.4 (7.0 for TX Matrix router)
Multilink Point-to-Point Protocol (MLPPP) and Multilink Frame Relay (MLFR)	7.0
Automatic Protection Switching (APS)—The current active interface (either the designated working or the designated protect interface) remains the active interface during a Routing Engine switchover.	7.4
Point-to-multipoint Multiprotocol Label Switching MPLS LSPs (transit only)	7.4
Compressed Real-Time Transport Protocol (CRTP)	7.6
Virtual private LAN service (VPLS)	8.2
Ethernet Operation, Administration, and Management (OAM) as defined by IEEE 802.3ah	8.5
Extended DHCP relay agent	8.5
Ethernet OAM as defined by IEEE 802.1ag	9.0
Packet Gateway Control Protocol (PGCP) process (pgcpd) on Multiservices 500 PICs on T640 routers.	9.0
Subscriber access	9.4
Layer 2 Circuit and LDP-based VPLS pseudowire redundant configuration	9.6

The following constraints apply to graceful Routing Engine switchover feature support:

- When graceful Routing Engine switchover and aggregated Ethernet interfaces are configured in the same system, the aggregated Ethernet interfaces must not be configured for fast-polling LACP. When fast polling is configured, the LACP polls time out at the remote end during the Routing Engine mastership switchover. When LACP polling times out, the aggregated link and interface are disabled. The Routing Engine mastership change is fast enough that standard and slow LACP polling do not time out during the procedure. However, note that this restriction does not apply to MX Series Routers that are running Junos OS Release 9.4 or later and have distributed periodic packet management (PPM) enabled—which is the default configuration—on them. In such cases, you can configure graceful Routing Engine switchover and have aggregated Ethernet interfaces configured for fast-polling LACP on the same device.

- VRRP changes mastership when a Routing Engine switchover occurs, even when graceful Routing Engine switchover is configured.

Graceful Routing Engine Switchover DPC Support

Graceful Routing Engine switchover supports all Dense Port Concentrators (DPCs) on the MX Series 3D Universal Edge Routers running the appropriate version of Junos OS. For more information about DPCs, see the *MX Series DPC Guide*.

Graceful Routing Engine Switchover and Subscriber Access

Graceful Routing Engine switchover currently supports most of the features directly associated with dynamic DHCP and dynamic PPPoE subscriber access. Graceful Routing Engine switchover also supports the unified in-service software upgrade (ISSU) for the DHCP access model and the PPPoE access model used by subscriber access.

Graceful Routing Engine Switchover PIC Support

Graceful Routing Engine switchover is supported on most PICs, except for the services PICs listed in this section. The PIC must be on a supported routing platform running the appropriate version of Junos OS. For information about FPC types, FPC/PIC compatibility, and the initial Junos OS Release in which an FPC supported a particular PIC, see the PIC guide for your router platform.

The following constraints apply to graceful Routing Engine switchover support for services PICs:

- You can include the **graceful-switchover** statement at the **[edit chassis redundancy]** hierarchy level on a router with Adaptive Services, Multiservices, and Tunnel Services PICs configured on it and successfully commit the configuration. However, all services on these PICs—except the Layer 2 service packages and extension-provider and SDK applications on Multiservices PICs—are reset during a switchover.
- Graceful Routing Engine switchover is not supported on any Monitoring Services PICs or Multilink Services PICs. If you include the **graceful-switchover** statement at the **[edit chassis redundancy]** hierarchy level on a router with either of these PIC types configured on it and issue the **commit** command, the commit fails.
- Graceful Routing Engine switchover is not supported on Multiservices 400 PICs configured for monitoring services applications. If you include the **graceful-switchover** statement, the commit fails.

NOTE: When an unsupported PIC is online, you cannot enable graceful Routing Engine switchover. If graceful Routing Engine switchover is already enabled, an unsupported PIC cannot come online.

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Understanding Graceful Routing Engine Switchover in the Junos OS on page 51
- Configuring Graceful Routing Engine Switchover on page 59
- Requirements for Routers with a Backup Router Configuration on page 60

CHAPTER 6

Graceful Routing Engine Switchover Configuration Guidelines

This chapter contains the following information:

Configuring Graceful Routing Engine Switchover

This section contains the following topics:

Enabling Graceful Routing Engine Switchover

By default, graceful Routing Engine switchover is disabled. To configure graceful Routing Engine switchover, include the **graceful-switchover** statement at the **[edit chassis redundancy]** hierarchy level.

```
[edit chassis redundancy]
graceful-switchover;
```

When you enable graceful Routing Engine switchover, the command-line interface (CLI) indicates which Routing Engine you are using. For example:

```
{master} [edit]
user@host#
```

To disable graceful Routing Engine switchover, delete the **graceful-switchover** statement from the **[edit chassis redundancy]** hierarchy level.

Synchronizing the Routing Engine Configuration

NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the master Routing Engine configuration.

When you configure graceful Routing Engine switchover, you can bring the backup Routing Engine online after the master Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

Verifying Graceful Routing Engine Switchover Operation

To verify whether graceful Routing Engine switchover is enabled, on the backup Routing Engine, issue the **show system switchover** command. When the output of the command indicates that the **Graceful switchover** field is set to **on**, graceful Routing Engine switchover is operational. The status of the kernel database and configuration database synchronization between Routing Engines is also provided. For example:

```
Graceful switchover: On
Configuration database: Ready
Kernel database: Ready
Peer state: Steady state
```

NOTE: You must issue the **show system switchover** command on the backup Routing Engine. This command is not supported on the master Routing Engine.

For more information about the **show system switchover** command, see the Junos OS Operational Mode Commands.

Related Documentation

- Understanding Graceful Routing Engine Switchover in the Junos OS on page 51
- Graceful Routing Engine Switchover System Requirements on page 55
- Requirements for Routers with a Backup Router Configuration on page 60
- Resetting Local Statistics on page 61
- **graceful-switchover on page 63**

Requirements for Routers with a Backup Router Configuration

If your Routing Engine configuration includes a **backup-router** statement or an **inet6-backup-router** statement, you can also use the **destination** statement to specify a subnet address or multiple subnet addresses for the backup router. Include destination subnets for the backup Routing Engine at the **[edit system (backup-router | inet6-backup-router) address]** hierarchy level. This requirement also applies to any T640 router connected to a TX Matrix router that includes a **backup-router** or **inet6-backup-router** statement.

NOTE: If you have a backup router configuration in which multiple static routes point to a gateway from the management Ethernet interface, you must configure prefixes that are more specific than the static routes or include the retain flag at the **[edit routing-options static route]** hierarchy level.

For example, if you configure the static route 172.16.0.0/12 from the management Ethernet interface for management purposes, you must specify the backup router configuration as follows:

```
backup-router 172.29.201.62 destination [172.16.0.0/13 172.16.128.0/13]
```

Related Documentation

- Understanding Graceful Routing Engine Switchover in the Junos OS on page 51
- Graceful Routing Engine Switchover System Requirements on page 55

Resetting Local Statistics

When you enable graceful Routing Engine switchover, the master Routing Engine configuration is copied and loaded to the backup Routing Engine. User files, accounting information, and trace options information are not replicated to the backup Routing Engine.

When a graceful Routing Engine switchover occurs, local statistics such as process statistics and networking statistics are displayed as a cumulative value from the time the process first came online. Because processes on the master Routing Engine can start at different times from the processes on the backup Routing Engine, the statistics on the two Routing Engines for the same process might differ. After a graceful Routing Engine switchover, we recommend that you issue the **clear interface statistics (*interface-name* | all)** command to reset the cumulative values for local statistics. Forwarding statistics are not affected by graceful Routing Engine switchover.

For information about how to use the **clear** command to clear statistics and protocol database information, see the Junos OS Operational Mode Commands.

NOTE: The **clear firewall** command cannot be used to clear the Routing Engine filter counters on a backup Routing Engine that is enabled for graceful Routing Engine switchover.

Related Documentation

- Understanding Graceful Routing Engine Switchover in the Junos OS on page 51
- Configuring Graceful Routing Engine Switchover on page 59

CHAPTER 7

Summary of Graceful Routing Engine Switchover Configuration Statements

This chapter provides a reference for the graceful Routing Engine switchover configuration statement.

graceful-switchover

Syntax graceful-switchover;

Hierarchy Level [edit chassis redundancy]

Release Information Statement introduced before Junos OS Release 7.4.

Description For routing platforms with two Routing Engines, configure a master Routing Engine to switch over gracefully to a backup Routing Engine without interruption to packet forwarding.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Routing Engine Switchover on page 59

PART 4

Nonstop Bridging

CHAPTER 8

Nonstop Bridging Overview

This chapter contains the following information:

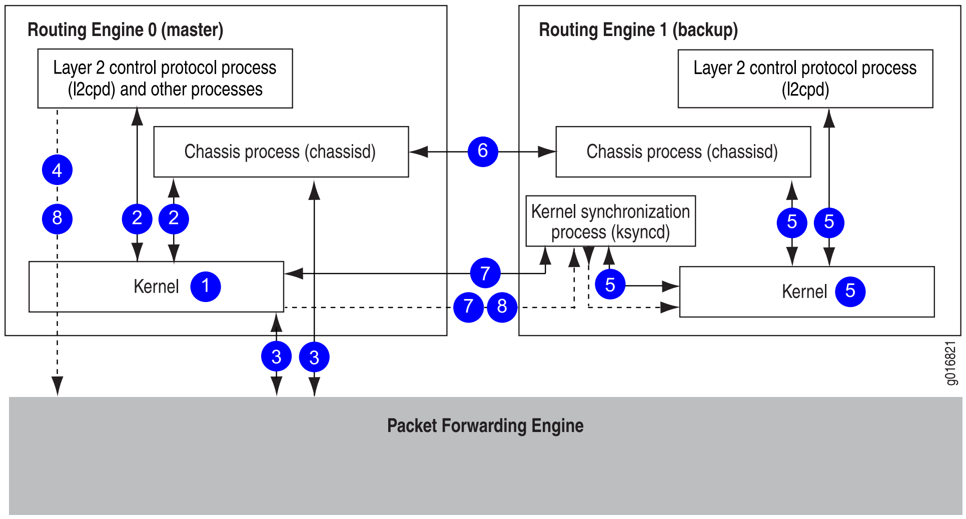
Nonstop Bridging Concepts

Nonstop bridging uses the same infrastructure as graceful Routing Engine switchover (GRES) to preserve interface and kernel information. However, nonstop bridging also saves Layer 2 Control Protocol (L2CP) information by running the Layer 2 Control Protocol process (l2cpd) on the backup Routing Engine.

NOTE: To use nonstop bridging, you must first enable graceful Routing Engine switchover on your routing platform. For more information about graceful Routing Engine switchover, see "Understanding Graceful Routing Engine Switchover in the Junos OS" on page 51.

Figure 3 on page 68 shows the system architecture of nonstop bridging and the process a routing platform follows to prepare for a switchover.

Figure 3: Nonstop Bridging Switchover Preparation Process

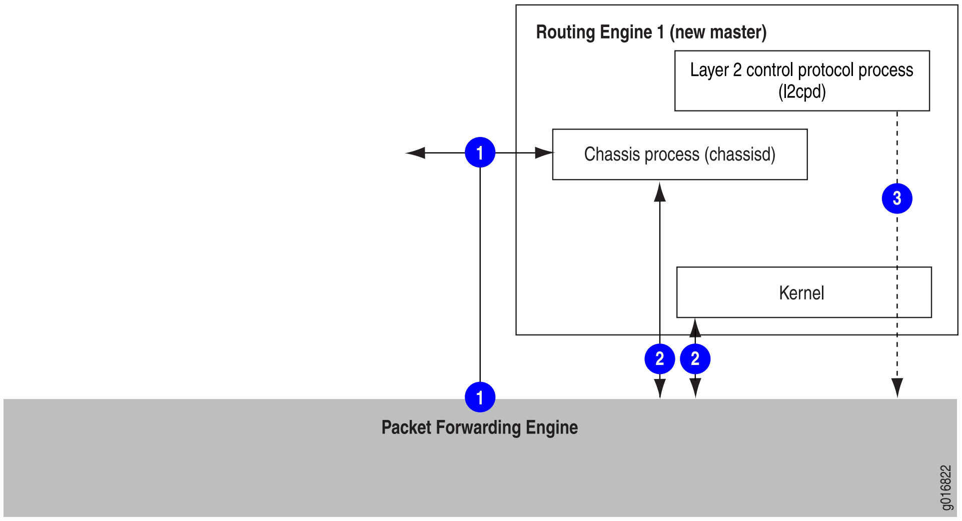


The switchover preparation process for nonstop bridging follows these steps:

1. The master Routing Engine starts.
2. The routing platform processes on the master Routing Engine (such as the chassis process [chassisd] and the Layer 2 Control Protocol process [l2cpd]) start.
3. The Packet Forwarding Engine starts and connects to the master Routing Engine.
4. All state information is updated in the system.
5. The backup Routing Engine starts, including the chassis process (chassisd) and the Layer 2 Control Protocol process (l2cpd).
6. The system determines whether graceful Routing Engine switchover and nonstop bridging have been enabled.
7. The kernel synchronization process (ksyncd) synchronizes the backup Routing Engine with the master Routing Engine.
8. For supported protocols, state information is updated directly between the l2cpds on the master and backup Routing Engines.

Figure 4 on page 69 shows the effects of a switchover on the routing platform.

Figure 4: Nonstop Bridging During a Switchover



The switchover process follows these steps:

1. When keepalives from the master Routing Engine are lost, the system switches over gracefully to the backup Routing Engine.
2. The Packet Forwarding Engine connects to the backup Routing Engine, which becomes the new master. Because the Layer 2 Control Protocol process (l2cpd) and chassis process (chassisd) are already running, these processes do not need to restart.
3. State information learned from the point of the switchover is updated in the system. Forwarding and bridging are continued during the switchover, resulting in minimal packet loss.

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Nonstop Bridging System Requirements on page 69
- Configuring Nonstop Bridging on page 71

- Configuring Nonstop Bridging on EX Series Switches (CLI Procedure)

Nonstop Bridging System Requirements

This topic contains the following sections:

- Platform Support on page 69
- Protocol Support on page 70

Platform Support

Nonstop bridging is supported on MX Series 3D Universal Edge Routers. Your system must be running Junos OS Release 8.4 or later.

Nonstop bridging is supported on EX Series Ethernet Switch with redundant Routing Engines.

For a list of the EX Series switches and Layer 2 protocols that support nonstop bridging, see EX Series Switch Software Features Overview.

NOTE: **All Routing Engines configured for nonstop bridging must be running the same Junos OS release.**

Protocol Support

Nonstop bridging is supported for the following Layer 2 control protocols:

- Spanning Tree Protocol (STP)
- Rapid Spanning Tree Protocol (RSTP)
- Multiple Spanning Tree Protocol (MSTP)

Related Documentation

- Nonstop Bridging Concepts on page 67
- Configuring Nonstop Bridging on page 71
- Configuring Nonstop Bridging on EX Series Switches (CLI Procedure)

CHAPTER 9

Nonstop Bridging Configuration Guidelines

This chapter includes the following topic:

Configuring Nonstop Bridging

This section includes the following topics:

Enabling Nonstop Bridging

Nonstop bridging requires you to configure graceful Routing Engine switchover (GRES). To enable graceful Routing Engine switchover, include the **graceful-switchover** statement at the **[edit chassis redundancy]** hierarchy level:

```
[edit chassis redundancy]
graceful-switchover;
```

By default, nonstop bridging is disabled. To enable nonstop bridging, include the **nonstop-bridging** statement at the **[edit protocols layer2-control]** hierarchy level:

```
[edit protocols layer2-control]
nonstop-bridging;
```

To disable nonstop active routing, remove the **nonstop-bridging** statement from the **[edit protocols layer2–control]** hierarchy level.

Synchronizing the Routing Engine Configuration

When you configure nonstop bridging, you must also include the **commit synchronize** statement at the **[edit system]** hierarchy level so that, by default, when you issue the **commit** command, the configuration changes are synchronized on both Routing Engines. If you issue the **commit synchronize** command at the **[edit]** hierarchy level on the backup Routing Engine, the Junos OS displays a warning and commits the candidate configuration.

NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the master Routing Engine configuration.

When you configure nonstop bridging, you can bring the backup Routing Engine online after the master Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

Verifying Nonstop Bridging Operation

When you enable nonstop bridging, you can issue Layer 2 Control Protocol-related operational mode commands on the backup Routing Engine. However, the output of the commands might not match the output of the same commands issued on the master Routing Engine.

Related Documentation

- Nonstop Bridging Concepts on page 67
- Nonstop Bridging System Requirements on page 69
- **nonstop-bridging on page 73**
- Configuring Nonstop Bridging on EX Series Switches (CLI Procedure)

CHAPTER 10

Summary of Nonstop Bridging Statements

This chapter provides a reference for the **nonstop-bridging** configuration statement.

nonstop-bridging

Syntax nonstop-bridging;

Hierarchy Level [edit protocols layer2-control]

Release Information Statement introduced in Junos OS Release 8.4.

Description For routing platforms with two Routing Engines, configure a master Routing Engine to switch over gracefully to a backup Routing Engine and preserve Layer 2 Control Protocol (L2CP) information.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Synchronizing the Routing Engine Configuration on page 92
- Configuring Nonstop Bridging on page 71
- Configuring Nonstop Bridging on EX Series Switches (CLI Procedure)

PART 5

Nonstop Active Routing

CHAPTER 11

Nonstop Active Routing Overview

This chapter contains the following topics:

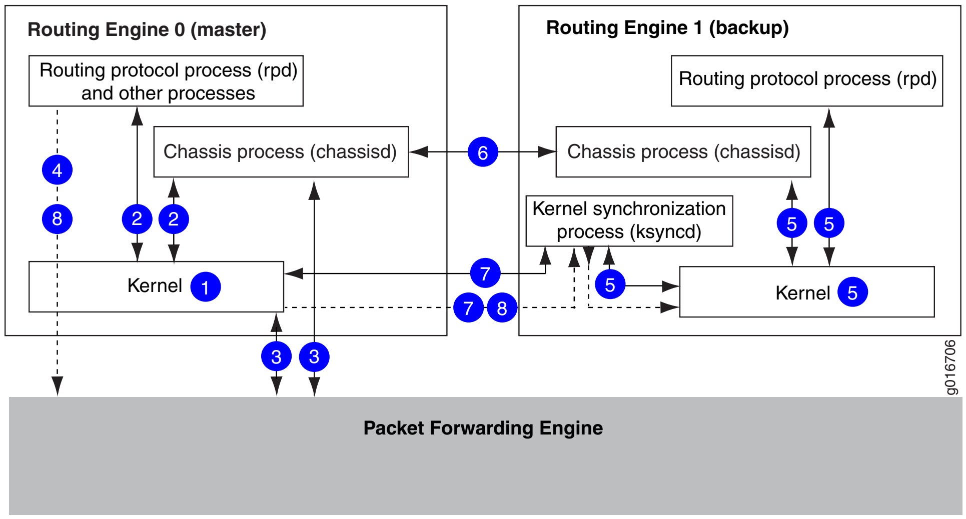
Nonstop Active Routing Concepts

Nonstop active routing (NSR) uses the same infrastructure as graceful Routing Engine switchover (GRES) to preserve interface and kernel information. However, nonstop active routing also saves routing protocol information by running the routing protocol process (rpd) on the backup Routing Engine. By saving this additional information, nonstop active routing is self-contained and does not rely on helper routers to assist the routing platform in restoring routing protocol information. Nonstop active routing is advantageous in networks where neighbor routers do not support graceful restart protocol extensions. As a result of this enhanced functionality, nonstop active routing is a natural replacement for graceful restart.

NOTE: To use nonstop active routing, you must first enable graceful Routing Engine switchover on your routing platform. For more information about graceful Routing Engine switchover, see "Understanding Graceful Routing Engine Switchover in the Junos OS" on page 51.

Figure 5 on page 78 shows the system architecture of nonstop active routing and the process a routing platform follows to prepare for a switchover.

Figure 5: Nonstop Active Routing Switchover Preparation Process

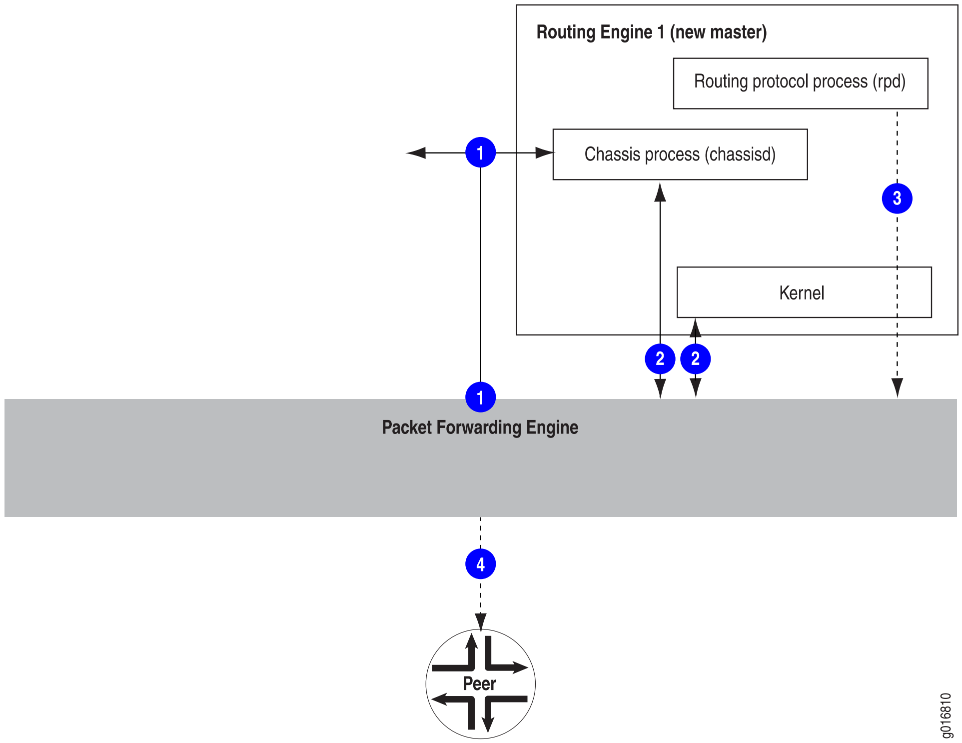


The switchover preparation process for nonstop active routing follows these steps:

1. The master Routing Engine starts.
2. The routing platform processes on the master Routing Engine (such as the chassis process [chassisd] and the routing protocol process [rpd]) start.
3. The Packet Forwarding Engine starts and connects to the master Routing Engine.
4. All state information is updated in the system.
5. The backup Routing Engine starts, including the chassis process (chassisd) and the routing protocol process (rpd).
6. The system determines whether graceful Routing Engine switchover and nonstop active routing have been enabled.
7. The kernel synchronization process (ksyncd) synchronizes the backup Routing Engine with the master Routing Engine.
8. For supported protocols, state information is updated directly between the routing protocol processes on the master and backup Routing Engines.

Figure 6 on page 79 shows the effects of a switchover on the routing platform.

Figure 6: Nonstop Active Routing During a Switchover



The switchover process follows these steps:

1. When keepalives from the master Routing Engine are lost, the system switches over gracefully to the backup Routing Engine.
2. The Packet Forwarding Engine connects to the backup Routing Engine, which becomes the new master. Because the routing protocol process (rpd) and chassis process (chassisd) are already running, these processes do not need to restart.
3. State information learned from the point of the switchover is updated in the system. Forwarding and routing are continued during the switchover, resulting in minimal packet loss.
4. Peer routers continue to interact with the routing platform as if no change had occurred. Routing adjacencies and session state relying on underlying routing information are preserved and not reset.

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Nonstop Active Routing System Requirements on page 80
- Configuring Nonstop Active Routing on page 91

Nonstop Active Routing System Requirements

This section contains the following topics:

Nonstop Active Routing Platform Support

Table 6 on page 80 lists the platforms that support nonstop active routing (NSR).

Table 6: Nonstop Active Routing Platform Support

Platform	Junos OS Release
M10i router	8.4 or later
M20 router	8.4 or later
M40e router	8.4 or later
M120 router	9.0 or later
M320 router	8.4 or later
MX Series routers	9.0 or later
PTX Series Packet Transport switches NOTE: Nonstop active routing (NSR) switchover on PTX series is supported only for the following MPLS and VPN protocols and applications using chained composite next hops: • Labeled BGP • Layer 2 VPNs excluding Layer 2 interworking (Layer 2 stitching) • Layer 3 VPNs • LDP • RSVP	12.1R4 or later
T320 router, T640 router, and TX Matrix router	8.4 or later
T1600 router	8.5 or later

Table 6: Nonstop Active Routing Platform Support *(continued)*

Platform	Junos OS Release
TX Plus Matrix router	10.0 or later

NOTE: All Routing Engines configured for nonstop active routing must be running the same Junos OS release.

Nonstop Active Routing Protocol and Feature Support

Table 7 on page 81 lists the protocols that are supported by nonstop active routing.

Table 7: Nonstop Active Routing Protocol and Feature Support

Protocol	Junos OS Release
Aggregated Ethernet interfaces with Link Aggregation Control Protocol (LACP)	9.4 or later
Bidirectional Forwarding Detection (BFD) For more information, see "Nonstop Active Routing BFD Support" on page 83.	8.5 or later
BGP For more information, see "Nonstop Active Routing BGP Support" on page 84.	8.4 or later
Labeled BGP (PTX Series Packet Transport Switches only)	12.1R4 or later
IS-IS	8.4 or later
LDP	8.4 or later
LDP-based virtual private LAN service (VPLS)	9.3 or later
LDP OAM (operation, administration, and management) features	9.6 or later

Table 7: Nonstop Active Routing Protocol and Feature Support *(continued)*

Protocol	Junos OS Release
LDP (PTX Series Packet Transport Switches only) Nonstop active routing support for LDP includes: • LDP unicast transit LSPs • LDP egress LSPs for labeled internal BGP (IBGP) and external BGP (EBGP) • LDP over RSVP transit LSPs • LDP transit LSPs with indexed next hops • LDP transit LSPs with unequal cost load balancing NOTE: Nonstop active routing is not supported for LDP Point-to-Multipoint LSPs and LDP ingress LSPs.	12.1R4 or later
Layer 2 circuits	(on LDP-based VPLS) 9.2 or later (on RSVP-TE LSP) 11.1 or later
Layer 2 VPNs	9.1 or later
Layer 2 VPNs (PTX Series Packet Transport Switches only) NOTE: Nonstop active routing is not supported for Layer 2 interworking (Layer 2 stitching).	12.1R4 or later
Layer 3 VPNs (see the first Note after this table for restrictions)	9.2 or later
Layer 3 VPNs (PTX Series Packet Transport Switches only)	12.1R4 or later
Multicast Source Discovery Protocol (MSDP) For more information, see “Nonstop Active Routing MSDP Support” on page 88.	12.1 or later
OSPF/OSPFv3	8.4 or later
Protocol Independent Multicast (PIM) For more information, see “Nonstop Active Routing PIM Support” on page 85.	(for IPv4) 9.3 or later (for IPv6) 10.4 or later
RIP and RIP next generation (RIPng)	9.0 or later

Table 7: Nonstop Active Routing Protocol and Feature Support *(continued)*

Protocol	Junos OS Release

RSVP (PTX Series Packet Transport Switches only)
 	
Nonstop active routing support for RSVP includes:
 	
• Point-to-Multipoint LSPs
  • RSVP Point-to-Multipoint ingress, transit, and egress LSPs using existing non-chained next hop.
  • RSVP Point-to-Multipoint transit LSPs using composite next hops for Point-to-Multipoint label routes.
• Point-to-Point LSPs
  • RSVP Point-to-Point ingress, transit, and egress LSPs using non-chained next hops.
  • RSVP Point-to-Point transit LSPs using chained composite next hops.
	12.1R4 or later

RSVP-TE LSP For more information, see "Nonstop Active Routing Support for RSVP-TE LSPs" on page 89.	9.5 or later
VPLS	(LDP-based) 9.1 or later (RSVP-TE-based) 11.2 or later

NOTE: Layer 3 VPN support does not include dynamic GRE tunnels, multicast VPNs, or BGP flow routes.

NOTE: If you configure a protocol that is not supported by nonstop active routing, the protocol operates as usual. When a switchover occurs, the state information for the unsupported protocol is not preserved and must be refreshed using the normal recovery mechanisms inherent in the protocol.

NOTE: On routers that have logical systems configured on them, only the master logical system supports nonstop active routing.

Nonstop Active Routing BFD Support

Nonstop active routing supports the Bidirectional Forwarding Detection (BFD) protocol, which uses the topology discovered by routing protocols to monitor neighbors. The BFD protocol is a simple hello mechanism that detects failures in a network. Because BFD is streamlined to be efficient at fast liveness detection, when it is used in conjunction with routing protocols, routing recovery times are improved. With nonstop active routing enabled, BFD session states are not restarted when a Routing Engine switchover occurs.

NOTE: BFD session states are saved only for clients using aggregate or static routes or for BGP, IS-IS, OSPF/OSPFv3, or PIM.

When a BFD session is distributed to the Packet Forwarding Engine, BFD packets continue to be sent during a Routing Engine switchover. If nondistributed BFD sessions are to be kept alive during a switchover, you must ensure that the session failure detection time is greater than the Routing Engine switchover time. The following BFD sessions are not distributed to the Packet Forwarding Engine: multihop sessions, tunnel-encapsulated sessions, and sessions over integrated routing and bridging (IRB) interfaces.

NOTE: BFD is an intensive protocol that consumes system resources. Specifying a minimum interval for BFD less than 100 ms for Routing Engine-based sessions and 10 ms for distributed BFD sessions can cause undesired BFD flapping. The **minimum-interval** configuration statement is a BFD liveness detection parameter.

Depending on your network environment, these additional recommendations might apply:

- For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of 300 ms for Routing Engine-based sessions, and 100 ms for distributed BFD sessions.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with nonstop active routing configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

Nonstop Active Routing BGP Support

Nonstop active routing BGP support is subject to the following conditions:

- You must include the **path-selection external-router-ID** statement at the **[edit protocols bgp]** hierarchy level to ensure consistent path selection between the master and backup Routing Engines during and after the nonstop active routing switchover.
- If the BGP peer in the master Routing Engine has negotiated address-family capabilities that are not supported for nonstop active routing, then the corresponding BGP neighbor state on the backup Routing Engine shows as idle. On switchover, the BGP session is reestablished from the new master Routing Engine.

Only the following address families are supported for nonstop active routing.

NOTE: **Address families are supported only on the main instance of BGP. Only unicast is supported on VRF instances.**

- inet unicast
- inet labeled-unicast
- inet multicast
- inet6 labeled-unicast
- inet6 multicast
- inet6 unicast
- route-target
- l2vpn signaling
- inet6-vpn unicast
- inet-vpn unicast
- inet-mdt
- iso-vpn

- BGP route dampening does not work on the backup Routing Engine when nonstop active routing is enabled.

Nonstop Active Routing Layer 2 Circuit and VPLS Support

Nonstop active routing supports Layer 2 circuit and VPLS on both LDP-based and RSVP-TE-based networks. Nonstop active routing support enables the backup Routing Engine to track the label advertised by Layer 2 circuit and VPLS on the primary Routing Engine, and to use the same label after the Routing Engine switchover.

in Junos OS Release 9.6 and later, nonstop active routing support is extended to the Layer 2 circuit and LDP-based VPLS pseudowire redundant configurations.

Nonstop Active Routing PIM Support

Nonstop active routing supports Protocol Independent Multicast (PIM) with stateful replication on backup Routing Engines. State information replicated on the backup Routing Engine includes information about neighbor relationships, join and prune events, rendezvous point (RP) sets, synchronization between routes and next hops, multicast session states, and the forwarding state between the two Routing Engines.

NOTE: Nonstop active routing for PIM is supported for IPv4 on Junos OS Release 9.3 and later, and for IPv6 on Junos OS Release 10.4 and later. Starting with Release 11.1, Junos OS also supports nonstop active routing for PIM on devices that have both IPv4 and IPv6 configured on them.

To configure nonstop active routing for PIM, include the same statements in the configuration as for other protocols: the **nonstop-routing** statement at the **[edit routing-options]** hierarchy level and the **graceful-restart** statement at the **[edit chassis redundancy]** hierarchy level. To trace PIM nonstop active routing events, include the **flag nsr-synchronization** statement at the **[edit protocols pim traceoptions]** hierarchy level.

NOTE: The **clear pim join**, **clear pim register**, and **clear pim statistics** operational mode commands are not supported on the backup Routing Engine when nonstop active routing is enabled.

Nonstop active routing support varies for different PIM features. The features fall into the following three categories: supported features, unsupported features, and incompatible features.

Supported features:

- Auto-RP

  NOTE: Nonstop active routing PIM support on IPv6 does not support auto-RP because IPv6 does not support auto-RP.

- Bootstrap router (BSR)
- Static RPs
- Embedded RP on non-RP IPv6 routers
- Local RP

  NOTE: RP set information synchronization is supported for local RP and BSR (on IPv4 and IPv6), autoRP (on IPv4), and embedded RP (on IPv6).

- BFD
- Dense mode
- Sparse mode
- Source-specific multicast (SSM)
- Draft Rosen multicast VPNs (MVPNs)

- Anycast RP (anycast RP set information synchronization and anycast RP register state synchronization on IPv4 and IPv6 configurations)
- Flow maps
- Unified ISSU
- Policy features such as neighbor policy, bootstrap router export and import policies, scope policy, flow maps, and reverse path forwarding (RPF) check policies
- Upstream assert synchronization
- PIM join load balancing

Starting with Release 12.2, Junos OS extends the nonstop active routing PIM support to draft Rosen MVPNs. Nonstop active routing PIM support for draft Rosen MVPNs enables nonstop active routing-enabled devices to preserve draft Rosen MPVN-related information—such as default and data multicast distribution tree (MDT) states—across switchovers. In releases earlier than 12.2, nonstop active routing PIM configuration was incompatible with draft Rosen MVPN configuration.

The backup Routing Engine sets up the default MDT based on the configuration and the information it receives from the master Routing Engine, and keeps updating the default MDT state information.

However, for data MDTs, the backup Routing Engine relies on the master Routing Engine to provide updates when data MDTs are created, updated, or deleted. The backup Routing Engine neither monitors data MDT flow rates nor triggers a data MDT switchover based on variations in flow rates. Similarly, the backup Routing Engine does not maintain the data MDT delay timer or timeout timer. It does not send MDT join TLV packets for the data MDTs until it takes over as the master Routing Engine. After the switchover, the new master Routing Engine starts sending MDT join TLV packets for each data MDT, and also resets the data MDT timers. Note that the expiration time for the timers might vary from the original values on the previous master Routing Engine.

Starting with Release 12.3, Junos OS extends the Protocol Independent Multicast (PIM) nonstop active routing support to IGMP-only interfaces.

In Junos OS releases earlier than 12.3, the PIM joins created on IGMP-only interfaces were not replicated on the backup Routing Engine. Thus, the corresponding multicast routes were marked as pruned (meaning discarded) on the backup Routing Engine. Because of this limitation, after a switchover, the new master Routing Engine had to wait for the IGMP module to come up and start receiving reports to create PIM joins and to install multicast routes. This caused traffic loss until the multicast joins and routes were reinstated.

However, in Junos OS Release 12.3 and later, the multicast joins on the IGMP-only interfaces are mapped to PIM states, and these states are replicated on the backup Routing Engine. If the corresponding PIM states are available on the backup, the multicast routes are marked as forwarding on the backup Routing Engine. This enables uninterrupted traffic flow after a switchover. This enhancement covers IGMPv2, IGMPv3, MLDv1, and MLDv2 reports and leaves.

Unsupported features: You can configure the following PIM features on a router along with nonstop active routing, but they function as if nonstop active routing is not enabled. In other words, during Routing Engine switchover and other outages, their state information is not preserved, and traffic loss is to be expected.

- Internet Group Management Protocol (IGMP) exclude mode
- IGMP snooping

Incompatible features: Nonstop active routing does not support the following features, and you cannot configure them on a router enabled for PIM nonstop active routing. The commit operation fails if the configuration includes both nonstop active routing and one or more of these features:

- Next-generation MVPNs with PIM provider tunnels

Junos OS provides a configuration statement that disables nonstop active routing for PIM only, so that you can activate incompatible PIM features and continue to use nonstop active routing for the other protocols on the router. Before activating an incompatible PIM feature, include the **nonstop-routing disable** statement at the **[edit protocols pim]** hierarchy level. Note that in this case, nonstop active routing is disabled for all PIM features, not just incompatible features.

Nonstop Active Routing MSDP Support

Starting with Release 12.1, Junos OS extends nonstop active routing support to the Multicast Source Discovery Protocol (MSDP).

Nonstop active routing support for MSDP preserves the following MSDP-related information across the switchover:

- MSDP configuration and peer information
- MSDP peer socket information
- Source-active and related information

However, note that the following restrictions or limitations apply to nonstop active routing MSDP support:

- Because the backup Routing Engine learns the active source information by processing the source-active messages from the network, synchronizing of source active information between the master and backup Routing Engines might take up to 60 seconds. So, no planned switchover is allowed within 60 seconds of the initial replication of the sockets.
- Similarly, Junos OS does not support two planned switchovers within 240 seconds of each other.

Junos OS enables you to trace MSDP nonstop active routing events by including the **flag nsr-synchronization** statement at the **[edit protocols msdp traceoptions]** hierarchy level.

Nonstop Active Routing Support for RSVP-TE LSPs

Junos OS extends nonstop active routing support to label-switching routers (LSRs) and Layer 2 Circuits that are part of an RSVP-TE LSP. Nonstop active routing support on LSRs ensures that the master to backup Routing Engine switchover on an LSR remains transparent to the network neighbors and that the LSP information remains unaltered during and after the switchover.

You can use the **show rsvp version** command to view the nonstop active routing mode and state on an LSR. Similarly, you can use the **show mpls lsp** and **show rsvp session** commands on the standby Routing Engine to view the state recreated on the standby Routing Engine.

The Junos OS nonstop active routing feature is also supported on RSVP point-to-multipoint LSPs. Nonstop active routing support for RSVP point-to-multipoint egress and transit LSPs was added in Junos OS Release 11.4, and for ingress LSPs in Release 12.1. During the switchover, the LSP comes up on the backup Routing Engine that shares and synchronizes the state information with the master Routing Engine before and after the switchover. Nonstop active routing support for point-to-multipoint transit and egress LSPs ensures that the switchover remains transparent to the network neighbors, and preserves the LSP information across the switchover.

However, Junos OS nonstop active routing support for RSVP point-to-multipoint LSPs does not include support for dynamically created point-to-multipoint LSPs, such as VPLS and next-generation MVPNs.

The **show rsvp session detail** command enables you to check the point-to-multipoint LSP remerge state information (**P2MP LSP re-merge**; possible values are **head**, **member**, and **none**).

However, Junos OS does not support nonstop active routing for the following features:

- Generalized Multiprotocol Label Switching (GMPLS) and LSP hierarchy
- Interdomain or loose-hop expansion LSPs
- BFD liveness detection

Nonstop active routing support for RSVP-TE LSPs is subject to the following limitations and restrictions:

- Detour LSPs are not maintained across a switchover and so, detour LSPs might fail to come back online after the switchover.
- Control plane statistics corresponding to the **show rsvp statistics** and **show rsvp interface detail | extensive** commands are not maintained across Routing Engine switchovers.
- Statistics from the backup Routing Engine are not reported for **show mpls lsp statistics** and **monitor mpls label-switched-path** commands. However, if a switchover occurs, the backup Routing Engine, after taking over as the master, starts reporting statistics. Note that the **clear statistics** command issued on the old master Routing Engine does not have any effect on the new master Routing Engine, which reports statistics, including any uncleared statistics.

- State timeouts might take additional time during nonstop active routing switchover. For example, if a switchover occurs after a neighbor has missed sending two hello messages to the master, the new master Routing Engine waits for another three hello periods before timing out the neighbor.
- On the RSVP ingress router, if you configure auto-bandwidth functionality, the bandwidth adjustment timers are set in the new master after the switchover. This causes a one-time increase in the length of time required for the bandwidth adjustment after the switchover occurs.
- RSVP ingress LSPs that have BFD liveness detection enabled on them do not come up on the backup Routing Engine during the switchover. Such BFD-enabled LSPs have to be reestablished after the switchover.
- Backup LSPs —LSPs that are established between the point of local repair (PLR) and the merge point after a node or link failure—are not preserved during a Routing Engine switchover.
- When nonstop active routing is enabled, graceful restart is not supported. However, graceful restart helper mode is supported.

Related Documentation

- Nonstop Active Routing Concepts on page 77
- Configuring Nonstop Active Routing on page 91

CHAPTER 12

Nonstop Active Routing Configuration Guidelines

This chapter contains the following sections:

Configuring Nonstop Active Routing

This section includes the following topics:

Enabling Nonstop Active Routing

Nonstop active routing (NSR) requires you to configure graceful Routing Engine switchover (GRES). To enable graceful Routing Engine switchover, include the **graceful-switchover** statement at the **[edit chassis redundancy]** hierarchy level:

```
[edit chassis redundancy]
graceful-switchover;
```

By default, nonstop active routing is disabled. To enable nonstop active routing, include the **nonstop-routing** statement at the **[edit routing-options]** hierarchy level:

```
[edit routing-options]
nonstop-routing;
```

To disable nonstop active routing, remove the **nonstop-routing** statement from the **[edit routing-options]** hierarchy level.

NOTE: When you enable nonstop active routing, you cannot enable automatic route distinguishers for multicast VPN routing instances. Automatic route distinguishers are enabled by configuring the **route-distinguisher-id** statement at the **[edit routing-instances *instance-name*]** hierarchy level; for more information, see the Junos OS VPNs Configuration Guide.

To enable the routing platform to switch over to the backup Routing Engine when the routing protocol process (rpd) fails rapidly three times in succession, include the **other-routing-engine** statement at the **[edit system processes routing failover]** hierarchy level.

For more information about the **other-routing-engine** statement, see the Junos OS System Basics Configuration Guide.

Synchronizing the Routing Engine Configuration

When you configure nonstop active routing, you must also include the **commit synchronize** statement at the **[edit system]** hierarchy level so that configuration changes are synchronized on both Routing Engines:

```
[edit system]
commit synchronize;
```

If you try to commit the nonstop active routing configuration without including the **commit synchronize** statement, the commit fails.

If you configure the **commit synchronize** statement at the **[edit system]** hierarchy level and issue a commit in the master Routing Engine, the master configuration is automatically synchronized with the backup.

However, if the backup Routing Engine is down when you issue the commit, the Junos OS displays a warning and commits the candidate configuration in the master Routing Engine. When the backup Routing Engine comes up, its configuration will automatically be synchronized with the master.

NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the master Routing Engine configuration.

When you configure nonstop active routing, you can bring the backup Routing Engine online after the master Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

Verifying Nonstop Active Routing Operation

To see whether or not nonstop active routing is enabled, issue the **show task replication** command. For BGP nonstop active routing, you can also issue the **show bgp replication** command.

For more information about these commands, see the Junos OS Operational Mode Commands and Junos OS Operational Mode Commands, respectively.

When you enable nonstop active routing or graceful Routing Engine switchover and issue routing-related operational mode commands on the backup Routing Engine (such as **show route**, **show bgp neighbor**, **show ospf database**, and so on), the output might not match the output of the same commands issued on the master Routing Engine. For example, it is normal for the routing table on the backup Routing Engine to contain persistent phantom routes that are not present in the routing table on the master Routing Engine.

To display BFD state replication status, issue the **show bfd session** command. The **replicated** flag appears in the output for this command when a BFD session has been replicated to the backup Routing Engine. For more information, see the Junos OS Operational Mode Commands.

Related Documentation

- Nonstop Active Routing Concepts on page 77
- Nonstop Active Routing System Requirements on page 80
- Tracing Nonstop Active Routing Synchronization Events on page 94
- Resetting Local Statistics on page 95
- Example: Configuring Nonstop Active Routing on page 95
- **nonstop-routing on page 101**

Preventing Automatic Reestablishment of BGP Peer Sessions After NSR Switchovers

It is useful to prevent a BGP peer session from automatically being reestablished after a nonstop active routing (NSR) switchover when you have applied routing policies configured in the dynamic database. When NSR is enabled, the dynamic database is not synchronized with the backup Routing Engine. Therefore, when a switchover occurs, import and export policies configured in the dynamic database might no longer be available. For more information about configuring dynamic routing policies, see the Routing Policy Configuration Guide.

NOTE: The BGP established timers are not maintained across switchovers.

You can configure the routing device not to reestablish a BGP peer session after an NSR switchover either for a specified period or until you manually reestablish the session. Include the **idle-after-switch-over** statement at the **[edit protocols bgp]** hierarchy level:

```
idle-after-switch-over (forever | seconds);
```

For a list of hierarchy levels at which you can configure this statement, see the configuration statement summary for this statement.

For ***seconds***, specify a value from 1 through 4294967295. The BGP peer session is not reestablished until after the specified period. If you specify the **forever** option, the BGP peer session is not reestablished until you issue the **clear bgp neighbor** command.

Tracing Nonstop Active Routing Synchronization Events

To track the progress of nonstop active routing synchronization between Routing Engines, you can configure nonstop active routing trace options flags for each supported protocol and for BFD sessions and record these operations to a log file.

To configure nonstop active routing trace options for supported routing protocols, include the **nsr-synchronization** statement at the **[edit protocols *protocol-name* traceoptions flag]** hierarchy level and optionally specify one or more of the **detail**, **disable**, **receive**, and **send** options:

```
[edit protocols]
bgp {
  traceoptions {
    flag nsr-synchronization <detail> <disable> <receive> <send>;
  }
}
isis {
  traceoptions {
    flag nsr-synchronization <detail> <disable> <receive> <send>;
  }
}
ldp {
  traceoptions {
    flag nsr-synchronization <detail> <disable> <receive> <send>;
  }
}
mpls {
  traceoptions {
    flag nsr-synchronization;
    flag nsr-synchronization-detail;
  }
}
msdp {
  traceoptions {
    flag nsr-synchronization <detail> <disable> <receive> <send>;
  }
}
(ospf | ospf3) {
  traceoptions {
    flag nsr-synchronization <detail> <disable> <receive> <send>;
  }
}
rip {
  traceoptions {
    flag nsr-synchronization <detail> <disable> <receive> <send>;
  }
}
ripng {
  traceoptions {
    flag nsr-synchronization <detail> <disable> <receive> <send>;
```

```
    }
  }
  pim {
    traceoptions {
      flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
  }
```

To configure nonstop active routing trace options for BFD sessions, include the **nsr-synchronization** and **nsr-packet** statements at the **[edit protocols bfd traceoptions flag]** hierarchy level.

```
[edit protocols]
bfd {
  traceoptions {
    flag nsr-synchronization;
    flag nsr-packet;
  }
}
```

To trace the Layer 2 VPN signaling state replicated from routes advertised by BGP, include the **nsr-synchronization** statement at the **[edit routing-options traceoptions flag]** hierarchy level. This flag also traces the label and logical interface association that VPLS receives from the kernel replication state.

```
[edit routing-options]
traceoptions {
  flag nsr-synchronization;
}
```

Related Documentation

- Configuring Nonstop Active Routing on page 91
- Configuring Nonstop Active Routing on EX Series Switches (CLI Procedure)
- Example: Configuring Nonstop Active Routing on page 95
- Example: Configuring Nonstop Active Routing on EX Series Switches

Resetting Local Statistics

After a graceful Routing Engine switchover, we recommend that you issue the **clear interface statistics (*interface-name* | all)** command to reset the cumulative values for local statistics on the new master Routing Engine.

Related Documentation

- Configuring Nonstop Active Routing on page 91
- Tracing Nonstop Active Routing Synchronization Events on page 94

Example: Configuring Nonstop Active Routing

The following example enables graceful Routing Engine switchover, nonstop active routing, and nonstop active routing trace options for BGP, IS-IS, and OSPF.

```
[edit]
system commit {
```

```
  synchronize;
}
chassis {
  redundancy {
    graceful-switchover; # This enables graceful Routing Engine switchover on
      # the routing platform.
  }
}
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.0.1.1/30;
      }
      family iso;
    }
  }
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.1.1.1/30;
      }
      family iso;
    }
  }
  so-0/0/2 {
    unit 0 {
      family inet {
        address 10.2.1.1/30;
      }
      family iso;
    }
  }
  so-0/0/3 {
    unit 0 {
      family inet {
        address 10.3.1.1/30;
      }
      family iso;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 192.168.2.1/32;
      }
      family iso {
        address 49.0004.1921.6800.2001.00;
      }
    }
  }
}
routing-options {
  nonstop-routing; # This enables nonstop active routing on the routing platform.
  router-id 192.168.2.1;
  autonomous-system 65432;
```

```
}
protocols {
  bgp {
    traceoptions {
      flag nsr-synchronization detail; # This logs nonstop active routing
        # events for BGP.
    }
    local-address 192.168.2.1;
    group external-group {
      type external;
      export BGP_export;
      neighbor 192.168.1.1 {
        family inet {
          unicast;
        }
        peer-as 65103;
      }
    }
    group internal-group {
      type internal;
      neighbor 192.168.10.1;
      neighbor 192.168.11.1;
      neighbor 192.168.12.1;
    }
  }
  isis {
    traceoptions {
      flag nsr-synchronization detail; # This logs nonstop active routing events
        # for IS-IS.
    }
    interface all;
    interface fxp0.0 {
      disable;
    }
    interface lo0.0 {
      passive;
    }
  }
  ospf {
    traceoptions {
      flag nsr-synchronization detail; # This logs nonstop active routing events
        # for OSPF.
    }
    area 0.0.0.0 {
      interface all;
      interface fxp0.0 {
        disable;
      }
      interface lo0.0 {
        passive;
      }
    }
  }
}
policy-options {
  policy-statement BGP_export {
```

```
        term direct {
            from {
                protocol direct;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
}
```

Related Documentation

- Configuring Nonstop Active Routing on page 91
- Tracing Nonstop Active Routing Synchronization Events on page 94

CHAPTER 13

Summary of Nonstop Active Routing Configuration Statements

This chapter provides a reference for each of the nonstop active routing configuration statements. The statements are organized alphabetically.

commit synchronize

Syntax commit synchronize;

Hierarchy Level [edit system]

Release Information Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 10.4 for EX Series switches.

Description For devices with multiple Routing Engines only. Configure the **commit** command to automatically result in a **commit synchronize** action between dual Routing Engines within the same chassis. The Routing Engine on which you execute the **commit** command (the requesting Routing Engine) copies and loads its candidate configuration to the other (the responding) Routing Engine. Each Routing Engine then performs a syntax check on the candidate configuration file being committed. If no errors are found, the configuration is activated and becomes the current operational configuration on both Routing Engines.

NOTE: **When you configure nonstop active routing (NSR), you must include the commit synchronize statement. Otherwise, the commit operation fails.**

On the TX Matrix router, synchronization only occurs between the Routing Engines within the same chassis and when synchronization is complete, the new configuration is then distributed to the Routing Engines on the T640 routers. That is, the master Routing Engine on the TX Matrix router distributes the configuration to the master Routing Engine on each T640 router. Likewise, the backup Routing Engine on the TX Matrix router distributes the configuration to the backup Routing Engine on each T640 router.

In EX Series Virtual Chassis configurations:

- On EX4200 switches in Virtual Chassis, synchronization occurs between the switch in the master role and the switch in the backup role.
- On EX8200 switches in a Virtual Chassis, synchronization occurs only between the master and backup XRE200 External Routing Engines.

Options **and-quit**—(Optional) (EX Series only) Quit configuration mode if the commit synchronization succeeds.

comment—(Optional) (EX Series only) Write a message to the commit log.

and-force—(Optional) (EX Series only) Force a commit synchronization on the other Routing Engine (ignore warnings).

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

Related Documentation

- Synchronizing the Routing Engine Configuration on page 92

- Configuring Multiple Routing Engines to Synchronize Committed Configurations Automatically

nonstop-routing

Syntax nonstop-routing;

Hierarchy Level [edit logical-systems *logical-system-name* routing-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options],
[edit routing-instances *routing-instance-name* routing-options],
[edit routing-options]

Release Information Statement introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 10.4 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description For routing platforms with two Routing Engines, configure a master Routing Engine to switch over gracefully to a backup Routing Engine and to preserve routing protocol information.

Default disabled

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Nonstop Active Routing on page 91
- Configuring Nonstop Active Routing on EX Series Switches (CLI Procedure)

PART 6

Graceful Restart

CHAPTER 14

Graceful Restart Overview

This chapter contains the following sections:

Graceful Restart Concepts

With routing protocols, any service interruption requires that an affected router recalculate adjacencies with neighboring routers, restore routing table entries, and update other protocol-specific information. An unprotected restart of a router can result in forwarding delays, route flapping, wait times stemming from protocol reconvergence, and even dropped packets. The main benefits of graceful restart are uninterrupted packet forwarding and temporary suppression of all routing protocol updates. Graceful restart enables a router to pass through intermediate convergence states that are hidden from the rest of the network.

Three main types of graceful restart are available on Juniper Networks routing platforms:

- Graceful restart for aggregate and static routes and for routing protocols—Provides protection for aggregate and static routes and for Border Gateway Protocol (BGP), End System-to-Intermediate System (ES-IS), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), next-generation RIP (RIPng), and Protocol Independent Multicast (PIM) sparse mode routing protocols.
- Graceful restart for MPLS-related protocols—Provides protection for Label Distribution Protocol (LDP), Resource Reservation Protocol (RSVP), circuit cross-connect (CCC), and translational cross-connect (TCC).
- Graceful restart for virtual private networks (VPNs)—Provides protection for Layer 2 and Layer 3 VPNs.

Graceful restart works similarly for routing protocols and MPLS protocols and combines components of these protocol types to enable graceful restart in VPNs. The main benefits of graceful restart are uninterrupted packet forwarding and temporary suppression of all routing protocol updates. Graceful restart thus enables a router to pass through intermediate convergence states that are hidden from the rest of the network.

Most graceful restart implementations define two types of routers—the restarting router and the helper router. The restarting router requires rapid restoration of forwarding state information so it can resume the forwarding of network traffic. The helper router assists the restarting router in this process. Graceful restart configuration statements typically affect either the restarting router or the helper router.

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Graceful Restart System Requirements on page 106
- Aggregate and Static Routes on page 107
- Graceful Restart and Routing Protocols on page 107
- Graceful Restart and MPLS-Related Protocols on page 110
- Graceful Restart and Layer 2 and Layer 3 VPNs on page 111
- Graceful Restart on Logical Systems on page 112
- Example: Configuring Graceful Restart on page 130
- Configuring Graceful Restart for QFabric Systems

Graceful Restart System Requirements

Graceful restart is supported on all routing platforms. To implement graceful restart for particular features, your system must meet these minimum requirements:

- Junos OS Release 5.3 or later for aggregate route, BGP, IS-IS, OSPF, RIP, RIPng, or static route graceful restart.
- Junos OS Release 5.5 or later for RSVP on egress provider edge (PE) routers.
- Junos OS Release 5.5 or later for LDP graceful restart.
- Junos OS Release 5.6 or later for the CCC, TCC, Layer 2 VPN, or Layer 3 VPN implementations of graceful restart.
- Junos OS Release 6.1 or later for RSVP graceful restart on ingress PE routers.
- Junos OS Release 6.4 or later for PIM sparse mode graceful restart.
- Junos OS Release 7.4 or later for ES-IS graceful restart (J Series Services Routers).
- Junos OS Release 8.5 or later for BFD session (helper mode only)—If a node is undergoing a graceful restart and its BFD sessions are distributed to the Packet Forwarding Engine, the peer node can help the peer with the graceful restart.
- Junos OS Release 9.2 or later for BGP to support helper mode without requiring that graceful restart be configured.

Related Documentation

- Graceful Restart Concepts on page 105

Aggregate and Static Routes

When you include the **graceful-restart** statement at the **[edit routing-options]** hierarchy level, any static routes or aggregated routes that have been configured are protected. Because no helper router assists in the restart, these routes are retained in the forwarding table while the router restarts (rather than being discarded or refreshed).

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Enabling Graceful Restart on page 113
- Verifying Graceful Restart Operation on page 128
- Example: Configuring Graceful Restart on page 130

Graceful Restart and Routing Protocols

This section covers the following topics:

BGP

When a router enabled for BGP graceful restart restarts, it retains BGP peer routes in its forwarding table and marks them as stale. However, it continues to forward traffic to other peers (or receiving peers) during the restart. To reestablish sessions, the restarting router sets the “restart state” bit in the BGP OPEN message and sends it to all participating peers. The receiving peers reply to the restarting router with messages containing end-of-routing-table markers. When the restarting router receives all replies from the receiving peers, the restarting router performs route selection, the forwarding table is updated, and the routes previously marked as stale are discarded. At this point, all BGP sessions are reestablished and the restarting peer can receive and process BGP messages as usual.

While the restarting router does its processing, the receiving peers also temporarily retain routing information. When a receiving peer detects a TCP transport reset, it retains the routes received and marks the routes as stale. After the session is reestablished with the restarting router, the stale routes are replaced with updated route information.

ES-IS

When graceful restart for ES-IS is enabled, the routes to end systems or intermediate systems are not removed from the forwarding table. The adjacencies are reestablished after restart is complete.

NOTE: ES-IS is supported only on the J Series Services Router.

IS-IS

Normally, IS-IS routers move neighbor adjacencies to the down state when changes occur. However, a router enabled for IS-IS graceful restart sends out Hello messages with the Restart Request (RR) bit set in a restart type length value (TLV) message. This indicates to neighboring routers that a graceful restart is in progress and to leave the IS-IS adjacency intact. The neighboring routers must interpret and implement restart signaling themselves. Besides maintaining the adjacency, the neighbors send complete sequence number PDUs (CSNPs) to the restarting router and flood their entire database.

The restarting router never floods any of its own link-state PDUs (LSPs), including pseudonode LSPs, to IS-IS neighbors while undergoing graceful restart. This enables neighbors to reestablish their adjacencies without transitioning to the down state and enables the restarting router to reinitiate a smooth database synchronization.

OSPF and OSPFv3

When a router enabled for OSPF graceful restart restarts, it retains routes learned before the restart in its forwarding table. The router does not allow new OSPF link-state advertisements (LSAs) to update the routing table. This router continues to forward traffic to other OSPF neighbors (or helper routers), and sends only a limited number of LSAs during the restart period. To reestablish OSPF adjacencies with neighbors, the restarting router must send a grace LSA to all neighbors. In response, the helper routers enter helper mode and send an acknowledgement back to the restarting router. If there are no topology changes, the helper routers continue to advertise LSAs as if the restarting router had remained in continuous OSPF operation.

When the restarting router receives replies from all the helper routers, the restarting router selects routes, updates the forwarding table, and discards the old routes. At this point, full OSPF adjacencies are reestablished and the restarting router receives and processes OSPF LSAs as usual. When the helper routers no longer receive grace LSAs from the restarting router or the topology of the network changes, the helper routers also resume normal operation.

NOTE: For more information about the standard helper mode implementation, see RFC 3623, *Graceful OSPF Restart*.

Starting with Release 11.3, Junos OS supports the restart signaling-based helper mode for OSPF graceful restart configurations. The helper modes, both standard and restart signaling-based, are enabled by default. In restart signaling-based helper mode implementations, the restarting router relays the restart status to its neighbors only after the restart is complete. When the restart is complete, the restarting router sends hello messages to its helper routers with the **restart signal** (**RS**) bit set in the hello packet header. When a helper router receives a hello packet with the **RS** bit set in the header, the helper router returns a hello message to the restarting router. The reply hello message from the helper router contains the **ResyncState** flag and the **ResyncTimeout** timer that enable the restarting router to keep track of the helper routers that are syncing up with it. When all helpers complete the synchronization, the restarting router exits the restart mode.

NOTE:

For more information about restart signaling-based graceful restart helper mode implementation, see RFC 4811, *OSPF Out-of-Band Link State Database (LSDB) Resynchronization*, RFC 4812, *OSPF Restart Signaling*, and RFC 4813, *OSPF Link-Local Signaling*.

Restart signaling-based graceful restart helper mode is not supported for OSPFv3 configurations.

PIM Sparse Mode

PIM sparse mode uses a mechanism called a *generation identifier* to indicate the need for graceful restart. Generation identifiers are included by default in PIM hello messages. An initial generation identifier is created by each PIM neighbor to establish device capabilities. When one of the PIM neighbors restarts, it sends a new generation identifier to its neighbors. All neighbors that support graceful restart and are connected by point-to-point links assist by sending multicast updates to the restarting neighbor.

The restart phase completes when either the PIM state becomes stable or when the restart interval timer expires. If the neighbors do not support graceful restart or connect to each other using multipoint interfaces, the restarting router uses the restart interval timer to define the restart period.

RIP and RIPng

When a router enabled for RIP graceful restart restarts, routes that have been configured are protected. Because no helper router assists in the restart, these routes are retained in the forwarding table while the router restarts (rather than being discarded or refreshed).

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Configuring Routing Protocols Graceful Restart on page 114
- Verifying Graceful Restart Operation on page 128
- Example: Configuring Graceful Restart on page 130

Graceful Restart and MPLS-Related Protocols

This section contains the following topics:

LDP

LDP graceful restart enables a router whose LDP control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. It also enables a router on which helper mode is enabled to assist a neighboring router that is attempting to restart LDP.

During session initialization, a router advertises its ability to perform LDP graceful restart or to take advantage of a neighbor performing LDP graceful restart by sending the graceful restart TLV. This TLV contains two fields relevant to LDP graceful restart: the reconnect time and the recovery time. The values of the reconnect and recovery times indicate the graceful restart capabilities supported by the router.

The reconnect time is configured in Junos OS as 60 seconds and is not user-configurable. The reconnect time is how long the helper router waits for the restarting router to establish a connection. If the connection is not established within the reconnect interval, graceful restart for the LDP session is terminated. The maximum reconnect time is 120 seconds and is not user-configurable. The maximum reconnect time is the maximum value that a helper router accepts from its restarting neighbor.

When a router discovers that a neighboring router is restarting, it waits until the end of the recovery time before attempting to reconnect. The recovery time is the length of time a router waits for LDP to restart gracefully. The recovery time period begins when an initialization message is sent or received. This time period is also typically the length of time that a neighboring router maintains its information about the restarting router, so it can continue to forward traffic.

You can configure LDP graceful restart both in the master instance for the LDP protocol and for a specific routing instance. You can disable graceful restart at the global level for all protocols, at the protocol level for LDP only, and for a specific routing instance only.

RSVP

RSVP graceful restart enables a router undergoing a restart to inform its adjacent neighbors of its condition. The restarting router requests a grace period from the neighbor or peer, which can then cooperate with the restarting router. The restarting router can still forward MPLS traffic during the restart period; convergence in the network is not disrupted. The restart is not visible to the rest of the network, and the restarting router is not removed from the network topology. RSVP graceful restart can be enabled on both transit routers and ingress routers. It is available for both point-to-point LSPs and point-to-multipoint LSPs.

CCC and TCC

CCC and TCC graceful restart enables Layer 2 connections between customer edge (CE) routers to restart gracefully. These Layer 2 connections are configured with the **remote-interface-switch** or **lsp-switch** statements. Because these CCC and TCC connections have an implicit dependency on RSVP LSPs, graceful restart for CCC and TCC uses the RSVP graceful restart capabilities.

RSVP graceful restart must be enabled on the provider edge (PE) routers and provider (P) routers to enable graceful restart for CCC and TCC. Also, because RSVP is used as the signaling protocol for signaling label information, the neighboring router must use helper mode to assist with the RSVP restart procedures.

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Configuring Graceful Restart for MPLS-Related Protocols on page 123
- Example: Configuring Graceful Restart on page 130

Graceful Restart and Layer 2 and Layer 3 VPNs

VPN graceful restart uses three types of restart functionality:

1. BGP graceful restart functionality is used on all PE-to-PE BGP sessions. This affects sessions carrying any service signaling data for network layer reachability information (NLRI), for example, an IPv4 VPN or Layer 2 VPN NLRI.
2. OSPF, IS-IS, LDP, or RSVP graceful restart functionality is used in all core routers. Routes added by these protocols are used to resolve Layer 2 and Layer 3 VPN NLRI.
3. Protocol restart functionality is used for any Layer 3 protocol (RIP, OSPF, LDP, and so on) used between the PE and customer edge (CE) routers. This does not apply to Layer 2 VPNs because Layer 2 protocols used between the CE and PE routers do not have graceful restart capabilities.

Before VPN graceful restart can work properly, all of the components must restart gracefully. In other words, the routers must preserve their forwarding states and request neighbors to continue forwarding to the router in case of a restart. If all of the conditions are satisfied, VPN graceful restart imposes the following rules on a restarting router:

- The router must wait to receive all BGP NLRI information from other PE routers before advertising routes to the CE routers.
- The router must wait for all protocols in all routing instances to converge (or complete the restart process) before it sends CE router information to other PE routers. In other words, the router must wait for all instance information (whether derived from local configuration or advertisements received from a remote peer) to be processed before it sends this information to other PE routers.

- The router must preserve all forwarding state in the **instance.mpls.0** tables until the new labels and transit routes are allocated and announced to other PE routers (and CE routers in a carrier-of-carriers scenario).

  If any condition is not met, VPN graceful restart does not succeed in providing uninterrupted forwarding between CE routers across the VPN infrastructure.

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Configuring Logical System Graceful Restart on page 127
- Verifying Graceful Restart Operation on page 128
- Example: Configuring Graceful Restart on page 130

Graceful Restart on Logical Systems

Graceful restart for a logical system functions much as graceful restart does in the main router. The only difference is the location of the **graceful-restart** statement:

- For a logical system, include the **graceful-restart** statement at the **[edit logical-systems *logical-system-name* routing-options]** hierarchy level.
- For a routing instance inside a logical system, include the **graceful-restart** statement at both the **[edit logical-systems *logical-system-name* routing-options]** and **[edit logical-systems *logical-system-name* routing-instances *instance-name* routing-options]** hierarchy levels.

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Configuring Logical System Graceful Restart on page 127
- Verifying Graceful Restart Operation on page 128
- Example: Configuring Graceful Restart on page 130

CHAPTER 15

Graceful Restart Configuration Guidelines

To implement graceful restart, you must perform the configuration tasks described in the following sections:

Enabling Graceful Restart

Graceful restart is disabled by default. You must configure graceful restart at the **[edit routing-options]** hierarchy level to enable the feature.

For graceful restart to function properly, graceful restart must be enabled on the **[edit routing-instance *instance-name* routing-options]** or **[edit routing-options]** hierarchy level.

For example:

```
routing-options {
  graceful-restart;
}
```

NOTE: **If you configure graceful restart after a BGP or LDP session has been established, the BGP or LDP session restarts and the peers negotiate graceful restart capabilities.**

To disable graceful restart, include the **disable** statement. To configure a time period for complete restart, include the **restart-duration** statement. You can specify a number between 120 and 900.

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

When you include the **graceful-restart** statement at the **[edit routing-options]** hierarchy level, graceful restart is also enabled for aggregate and static routes.

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Aggregate and Static Routes on page 107
- Example: Configuring Graceful Restart on page 130

Configuring Routing Protocols Graceful Restart

This topic includes the following sections:

Enabling Graceful Restart

By default, graceful restart is disabled. To enable graceful restart, include the **graceful-restart** statement at the **[edit routing-instance *instance-name* routing-options]** or **[edit routing-options]** hierarchy level.

For example:

```
routing-options {
  graceful-restart;
}
```

To configure the duration of the graceful restart period, include the **restart-duration** at the **[edit routing-options graceful-restart]** hierarchy level.

NOTE: Helper mode (the ability to assist a neighboring router attempting a graceful restart) is enabled by default when you start the routing platform, even if graceful restart is not enabled. You can disable helper mode on a per-protocol basis.

```
[edit]
routing-options {
```



```
    graceful-restart {
      disable;
      restart-duration seconds;
    }
  }
```

To disable graceful restart globally, include the **disable** statement at the **[edit routing-options graceful-restart]** hierarchy level.

When graceful restart is enabled for all routing protocols at the **[edit routing-options graceful-restart]** hierarchy level, you can disable graceful restart on a per-protocol basis.

NOTE: If you configure graceful restart after a BGP or LDP session has been established, the BGP or LDP session restarts and the peers negotiate graceful restart capabilities. Also, the BGP peer routing statistics are reset to zero.

Configuring Graceful Restart Options for BGP

To configure the duration of the BGP graceful restart period, include the **restart-time** statement at the **[edit protocols bgp graceful-restart]** hierarchy level. To set the length of time the router waits to receive messages from restarting neighbors before declaring them down, include the **stale-routes-time** statement at the **[edit protocols bgp graceful-restart]** hierarchy level.

```
[edit]
protocols {
  bgp {
    graceful-restart {
      disable;
      restart-time seconds;
      stale-routes-time seconds;
    }
  }
}
routing-options {
  graceful-restart;
}
```

To disable BGP graceful restart capability for all BGP sessions, include the **disable** statement at the **[edit protocols bgp graceful-restart]** hierarchy level.

NOTE: To set BGP graceful restart properties or disable them for a group, include the desired statements at the **[edit protocols bgp group *group-name* graceful-restart]** hierarchy level.

To set BGP graceful restart properties or disable them for a specific neighbor in a group, include the desired statements at the **[edit protocols bgp group *group-name* neighbor *ip-address* graceful-restart]** hierarchy level.

NOTE: Configuring graceful restart for BGP resets the BGP peer routing statistics to zero. Also, existing BGP sessions restart, and the peers negotiate graceful restart capabilities.

Configuring Graceful Restart Options for ES-IS

On J Series Services Routers, to configure the duration of the ES-IS graceful restart period, include the **restart-duration** statement at the **[edit protocols esis graceful-restart]** hierarchy level.

```
[edit]
protocols {
  esis {
    graceful-restart {
      disable;
      restart-duration seconds;
    }
  }
}
routing-options {
  graceful-restart;
}
```

To disable ES-IS graceful restart capability, include the **disable** statement at the **[edit protocols esis graceful-restart]** hierarchy level.

Configuring Graceful Restart Options for IS-IS

To configure the duration of the IS-IS graceful restart period, include the **restart-duration** statement at the **[edit protocols isis graceful-restart]** hierarchy level.

```
[edit]
protocols {
  isis {
    graceful-restart {
      disable;
      helper-disable;
      restart-duration seconds;
    }
  }
}
routing-options {
  graceful-restart;
}
```

To disable IS-IS graceful restart helper capability, include the **helper-disable** statement at the **[edit protocols isis graceful-restart]** hierarchy level. To disable IS-IS graceful restart capability, include the **disable** statement at the **[edit protocols isis graceful-restart]** hierarchy level.

NOTE: If you configure Bidirectional Forwarding Detection (BFD) and graceful restart for IS-IS, graceful restart might not work as expected.

NOTE: You can also track graceful restart events with the **traceoptions** statement at the **[edit protocols isis]** hierarchy level. For more information, see "Tracking Graceful Restart Events" on page 120.

Configuring Graceful Restart Options for OSPF and OSPFv3

To configure the duration of the OSPF/OSPFv3 graceful restart period, include the **restart-duration** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level. To specify the length of time for which the router notifies helper routers that it has completed graceful restart, include the **notify-duration** at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level. Strict OSPF link-state advertisement (LSA) checking results in the termination of graceful restart by a helping router. To disable strict LSA checking, include the **no-strict-lsa-checking** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level.

```
[edit]
protocols {
  ospf | ospfv3{
    graceful-restart {
      disable;
      helper-disable
      no-strict-lsa-checking;
      notify-duration seconds;
      restart-duration seconds;
    }
  }
}
routing-options {
  graceful-restart;
}
```

To disable OSPF/OSPFv3 graceful restart, include the **disable** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level.

Starting with Release 11.3, the Junos OS supports both the standard (based on RFC 3623, *Graceful OSPF Restart*) and the restart signaling-based (as specified in RFC 4811, RFC 4812, and RFC 4813) helper modes for OSPF version 2 graceful restart configurations. Both the standard and restart signaling-based helper modes are enabled by default. To disable the helper mode for OSPF version 2 graceful restart configurations, include the **helper-disable <both | restart-signaling | standard>** statement at the **[edit protocols ospf graceful-restart]** hierarchy level. Note that the last committed statement always takes precedence over the previous one.

```
[edit protocols ospf]
  graceful-restart {
    helper-disable <both | restart-signaling | standard>
  }
```

To reenable the helper mode, delete the **helper-disable** statement from the configuration by using the **delete protocols ospf graceful-restarthelper-disable <restart-signaling | standard | both>** command. In this case also, the last executed command takes precedence over the previous ones.

NOTE:

Restart signaling-based helper mode is not supported for OSPFv3 configurations. To disable helper mode for OSPFv3 configurations, include the helper-disable statement at the **[edit protocols ospfv3 graceful-restart]** hierarchy level.

TIP: You can also track graceful restart events with the **traceoptions** statement at the **[edit protocols (ospf | ospf3)]** hierarchy level. For more information, see "Tracking Graceful Restart Events" on page 120.

NOTE: You cannot enable OSPFv3 graceful restart between a routing platform running Junos OS Release 7.5 and earlier and a routing platform running Junos OS Release 7.6 or later. As a workaround, make sure both routing platforms use the same Junos OS version.

NOTE: If you configure BFD and graceful restart for OSPF, graceful restart might not work as expected.

Configuring Graceful Restart Options for RIP and RIPng

To configure the duration of the RIP or RIPng graceful restart period, include the **restart-time** statement at the **[edit protocols (rip | ripng) graceful-restart]** hierarchy level.

```
[edit]
protocols {
  (rip | ripng) {
    graceful-restart {
      disable;
      restart-time seconds;
    }
  }
}
routing-options {
  graceful-restart;
}
```

To disable RIP or RIPng graceful restart capability, include the **disable** statement at the **[edit protocols (rip | ripng) graceful-restart]** hierarchy level.

Configuring Graceful Restart Options for PIM Sparse Mode

PIM sparse mode continues to forward existing multicast packet streams during a graceful restart, but does not forward new streams until after the restart is complete. After a restart, the routing platform updates the forwarding state with any updates that were received from neighbors and occurred during the restart period. For example, the routing platform relearns the join and prune states of neighbors during the restart, but does not apply the changes to the forwarding table until after the restart.

PIM sparse mode-enabled routing platforms generate a unique 32-bit random number called a generation identifier. Generation identifiers are included by default in PIM hello messages, as specified in the IETF Internet draft *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*. When a routing platform receives PIM hellos containing generation identifiers on a point-to-point interface, Junos OS activates an algorithm that optimizes graceful restart.

Before PIM sparse mode graceful restart occurs, each routing platform creates a generation identifier and sends it to its multicast neighbors. If a PIM sparse mode-enabled routing platform restarts, it creates a new generation identifier and sends it to its neighbors. When a neighbor receives the new identifier, it resends multicast updates to the restarting router to allow it to exit graceful restart efficiently. The restart phase completes when either the PIM state becomes stable or when the restart interval timer expires.

If a routing platform does not support generation identifiers or if PIM is enabled on multipoint interfaces, the PIM sparse mode graceful restart algorithm does not activate, and a default restart timer is used as the restart mechanism.

To configure the duration of the PIM graceful restart period, include the **restart-duration** statement at the **[edit protocols pim graceful-restart]** hierarchy level:

```
[edit]
protocols {
  pim {
    graceful-restart {
      disable;
      restart-duration seconds;
    }
  }
}
routing-options {
  graceful-restart;
}
```

To disable PIM sparse mode graceful restart capability, include the **disable** statement at the **[edit protocols pim graceful-restart]** hierarchy level.

NOTE: Multicast forwarding can be interrupted in two ways. First, if the underlying routing protocol is unstable, multicast reverse-path-forwarding (RPF) checks can fail and cause an interruption. Second, because the forwarding table is not updated during the graceful restart period, new multicast streams are not forwarded until graceful restart is complete.

Tracking Graceful Restart Events

To track the progress of a graceful restart event, you can configure graceful restart trace options flags for IS-IS and OSPF/OSPFv3. To configure graceful restart trace options, include the **graceful-restart** statement at the **[edit protocols *protocol* traceoptions flag]** hierarchy level:

```
[edit protocols]
isis {
  traceoptions {
    flag graceful-restart;
  }
}
(ospf | ospf3) {
  traceoptions {
    flag graceful-restart;
  }
}
```

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Graceful Restart and Routing Protocols on page 107
- Verifying Graceful Restart Operation on page 128
- Example: Configuring Graceful Restart on page 130

Example: Managing Helper Modes for OSPF Graceful Restart

Requirements

M Series or T Series routers running Junos OS Release 11.4 or later.

Overview

Junos OS Release 11.4 extends OSPF graceful restart support to include restart signaling-based helper mode. Both standard (RFC 3623-based) and restart signaling-based helper modes are enabled by default, irrespective of the graceful-restart configuration status on the routing device.

Junos OS, however, enables you to choose between the helper modes with the **helper-disable <standard | restart-signaling | both>** statement.

Configuration

Both standard and restart signaling-based helper modes are enabled by default, irrespective of the graceful-restart configuration status on the routing device. Junos OS allows you to disable or enable the helper modes based on your requirements.

To configure the helper mode options for graceful restart:

1. To enable graceful restart, add the **graceful-restart** statement at the **[edit routing-options]** hierarchy level.

   ```
   [edit routing-options]
   user@host# set graceful-restart
   ```

   The helper modes, both standard and restart signaling-based, are enabled by default.

2. To disable one or both of the helper modes, add the **helper-disable <both | restart-signaling | standard>** statement at the **[edit protocols ospf graceful-restart]** hierarchy level.

   - To disable both standard and restart signaling-based helper modes:

     ```
     [edit protocols ospf graceful-restart]
     user@host# set helper-disable both
     ```

   - To disable only the restart signaling-based helper mode:

     ```
     [edit protocols ospf graceful-restart]
     user@host# set helper-disable restart-signaling
     ```

   - To disable only the standard helper mode:

     ```
     [edit protocols ospf graceful-restart]
     user@host# set helper-disable standard
     ```

   NOTE: You must commit the configuration before the change takes effect.

   The last committed statement always takes precedence over the previous one.

3. To enable one or both of the helper modes when the helper modes are disabled, delete the **helper-disable <both | restart-signaling | standard>** statement from the **[edit protocols ospf graceful-restart]** hierarchy level.

   - To enable both standard and restart signaling-based helper modes:

     ```
     [edit protocols ospf graceful-restart]
     user@host# delete helper-disable
     ```

   - To enable the restart signaling-based helper mode:

     ```
     [edit protocols ospf graceful-restart]
     user@host# delete helper-disable restart-signaling
     ```

- To enable the standard helper mode:

  [edit protocols ospf graceful-restart]
  user@host# **delete helper-disable standard**

NOTE: You must commit the configuration before the change takes effect.

The last committed statement always takes precedence over the previous one.

Verification

Confirm that the configuration is working properly.

- Verifying OSPF Graceful Restart and Helper Mode Configuration on page 122

Verifying OSPF Graceful Restart and Helper Mode Configuration

Purpose Verify the OSPF graceful restart and helper mode configuration on a router.

Action

- Enter the **run show ospf overview** command from configuration mode.

  `user@host#` **run show ospf overview**

```
~
~
~
Restart: Enabled
  Restart duration: 180 sec
  Restart grace period: 210 sec
  Graceful restart helper mode: Enabled
  Restart-signaling helper mode: Enabled
~
~
~
```

Meaning The output shows that graceful restart and both of the helper modes are enabled.

Related Documentation

- Understanding Restart Signaling-Based Helper Mode Support for OSPF Graceful Restart
- Tracing Restart Signaling-Based Helper Mode Events for OSPF Graceful Restart on page 122
- helper-disable (OSPF) on page 161

Tracing Restart Signaling-Based Helper Mode Events for OSPF Graceful Restart

Junos OS provides a tracing option to log restart signaling-based helper mode events for OSPF graceful restart. To enable tracing for restart signaling-based helper mode events, include the **traceoptions flag restart-signaling** statement at the **[edit protocols ospf]** hierarchy level.

To enable tracing for restart signaling-based events:

1. Create a log file for saving the log.

   [edit protocols ospf]
   user@host# **set traceoptions file *ospf-log***

   where ***ospf-log*** is the name of the log file.

2. Enable tracing for restart signaling-based helper mode events.

   [edit protocols ospf]
   user@host# **set traceoptions flag restart-signaling**

3. Commit the configuration.

   [edit protocols ospf]
   user@host# **commit**

The logs are saved to the ***ospf-log*** file in the **/var/log** folder.

Viewing the Log File

To view the restart signaling-based events from the log file, type:

```
user@host> file show /var/log/ospf-log | match "restart signaling"
Jun 25 14:44:08.890216 OSPF Restart Signaling: Start helper mode for nbr ip
14.19.3.2 id 10.10.10.1
Jun 25 14:44:11.358636 OSPF restart signaling: Received DBD with R bit set from
nbr ip=14.19.3.2 id=10.10.10.1. Start oob-resync.
Jun 25 14:44:11.380198 OSPF restart signaling: Received DBD with LR bit on from
nbr ip=14.19.3.2 id=10.10.10.1. Save its oob-resync capability 1
Jun 25 14:44:11.467200 OSPF restart signaling: nbr fsm for nbr ip=14.19.3.2
id=10.10.10.1 moving to state Full. Reset oob-resync parameters.
```

Related Documentation

- Understanding Restart Signaling-Based Helper Mode Support for OSPF Graceful Restart
- Example: Managing Helper Modes for OSPF Graceful Restart on page 120
- helper-disable (OSPF) on page 161

Configuring Graceful Restart for MPLS-Related Protocols

This section contains the following topics:

Configuring Graceful Restart Globally

To configure graceful restart globally for all MPLS-related protocols, include the **graceful-restart** statement at the **[edit routing-options]** hierarchy level. To configure the duration of the graceful restart period, include the **restart-duration** at the **[edit routing-options graceful-restart]** hierarchy level:

```
[edit]
routing-options {
  graceful-restart {
    disable;
    restart-duration seconds;
  }
}
```

To disable graceful restart globally, include the **disable** statement at the **[edit routing-options graceful-restart]** hierarchy level.

Configuring Graceful Restart Options for RSVP, CCC, and TCC

Because CCC and TCC rely on RSVP, you must modify these three protocols as a single group.

To configure how long the router retains the state of its RSVP neighbors while they undergo a graceful restart, include the **maximum-helper-recovery-time** statement at the **[edit protocols rsvp graceful-restart]** hierarchy level. This value is applied to all neighboring routers, so it should be based on the time required by the slowest RSVP neighbor to recover.

To configure the delay between when the router discovers that a neighboring router has gone down and when it declares the neighbor down, include the **maximum-helper-restart-time** statement at the **[edit protocols rsvp graceful-restart]** hierarchy level. This value is applied to all neighboring routers, so it should be based on the time required by the slowest RSVP neighbor to restart.

```
[edit]
protocols {
  rsvp {
    graceful-restart {
      disable;
      helper-disable;
      maximum-helper-recovery-time;
      maximum-helper-restart-time;
    }
  }
}
routing-options {
  graceful-restart;
}
```

To disable RSVP, CCC, and TCC graceful restart, include the **disable** statement at the **[edit protocols rsvp graceful-restart]** hierarchy level. To disable RSVP, CCC, and TCC helper capability, include the **helper-disable** statement at the **[edit protocols rsvp graceful-restart]** hierarchy level.

Configuring Graceful Restart Options for LDP

When configuring graceful restart for LDP, you can include the following optional statements at the **[edit protocols ldp graceful-restart]** hierarchy level:

```
[edit protocols ldp graceful-restart]
disable;
helper-disable;
maximum-neighbor-reconnect-time seconds;
maximum-neighbor-recovery-time seconds;
reconnect-time seconds;
recovery-time seconds;

[edit routing-options]
graceful-restart;
```

The statements have the following effects on the graceful restart process:

- To configure the length of time required to reestablish a session after a graceful restart, include the **reconnect-time** statement; the range is 30 through 300 seconds. To limit the maximum reconnect time allowed from a restarting neighbor router, include the **maximum-neighbor-reconnect-time** statement; the range is 30 through 300 seconds.
- To configure the length of time that helper routers are required to maintain the old forwarding state during a graceful restart, include the **recovery-time** statement; the range is 120 through 1800 seconds. On the helper router, you can configure a statement that overrides the request from the restarting router and sets the maximum length of time the helper router will maintain the old forwarding state. To configure this feature, include the **maximum-neighbor-recovery-time** statement; the range is 140 through 1900 seconds.

  NOTE: The value for the recovery-time and maximum-neighbor-recovery-time statements at the **[edit protocols ldp graceful-restart]** hierarchy level should be approximately 80 seconds longer than the value for the **restart-duration** statement at the **[edit routing-options graceful-restart]** hierarchy level. Otherwise, a warning message appears when you try to commit the configuration.

- To disable LDP graceful restart capability, include the **disable** statement. To disable LDP graceful restart helper capability, include the **helper-disable** statement.

Configuring VPN Graceful Restart

Graceful restart allows a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. Without graceful restart, a control plane restart disrupts any VPN services provided by the router. Graceful restart is supported on Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, and VPLS.

To implement graceful restart for a Layer 2 VPN or Layer 3 VPN, perform the configuration tasks described in the following sections:

- Configuring Graceful Restart Globally on page 126
- Configuring Graceful Restart for the Routing Instance on page 126

Configuring Graceful Restart Globally

To enable graceful restart, include the **graceful-restart** statement at the **[edit routing-options]** hierarchy level. To configure a global duration for the graceful restart period, include the **restart-duration** statement at the **[edit routing-options graceful-restart]** hierarchy level.

```
[edit]
routing-options {
  graceful-restart {
    disable;
    restart-duration seconds;
  }
}
```

To disable graceful restart globally, include the **disable** statement at the **[edit routing-options graceful-restart]** hierarchy level.

Configuring Graceful Restart for the Routing Instance

For Layer 3 VPNs only, you must also configure graceful restart for all routing and MPLS-related protocols within a routing instance by including the **graceful-restart** statement at the **[edit routing-instances *instance-name* routing-options]** hierarchy level. Because you can configure multi-instance BGP and multi-instance LDP, graceful restart for a carrier-of-carriers scenario is supported. To configure the duration of the graceful restart period for the routing instance, include the **restart-duration** statement at the **[edit routing-instances *instance-name* routing-options]**.

```
[edit]
routing-instances {
  instance-name {
    routing-options {
      graceful-restart {
        disable;
        restart-duration seconds;
      }
    }
  }
}
```

You can disable graceful restart for individual protocols with the **disable** statement at the **[edit routing-instances *instance-name* protocols *protocol-name* graceful-restart]** hierarchy level.

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Graceful Restart and Layer 2 and Layer 3 VPNs on page 111
- Verifying Graceful Restart Operation on page 128

- Example: Configuring Graceful Restart on page 130

Configuring Logical System Graceful Restart

Graceful restart for a logical system functions much as graceful restart does in the main router. The only difference is the location of the **graceful-restart** statement.

The following topics describe what to configure to implement graceful restart in a logical system:

- Enabling Graceful Restart Globally on page 127
- Configuring Graceful Restart for a Routing Instance on page 127

Enabling Graceful Restart Globally

To enable graceful restart in a logical system, include the **graceful-restart** statement at the **[edit logical-systems *logical-system-name* routing-options]** hierarchy level. To configure a global duration of the graceful restart period, include the **restart-duration** statement at the **[edit logical-systems *logical-system-name* routing-options graceful-restart]** hierarchy level.

```
[edit]
logical-systems {
  logical-system-name {
    routing-options {
      graceful-restart {
        disable;
        restart-duration seconds;
      }
    }
  }
}
```

To disable graceful restart globally, include the **disable** statement at the **[edit logical-systems *logical-system-name* routing-options graceful-restart]** hierarchy level.

Configuring Graceful Restart for a Routing Instance

For Layer 3 VPNs only, you must also configure graceful restart globally for a routing instance inside a logical system. To configure, include the **graceful-restart** statement at the **[edit logical-systems *logical-system-name* routing-instances *instance-name* routing-options]** hierarchy level. Because you can configure multi-instance BGP and multi-instance LDP, graceful restart for a carrier-of-carriers scenario is supported. To configure the duration of the graceful restart period for the routing instance, include the **restart-duration** statement at the **[edit logical-systems *logical-system-name* routing-instances *instance-name* routing-options]**.

```
[edit]
logical-systems {
  logical-system-name {
    routing-instances {
      instance-name {
        routing-options {
          graceful-restart {
```



```
                        disable;
                        restart-duration seconds;
                    }
                }
            }
        }
    }
}
```

To disable graceful restart for individual protocols with the **disable** statement at the **[edit logical-systems** ***logical-system-name*** **routing-instances** ***instance-name*** **protocols** ***protocol-name*** **graceful-restart]** hierarchy level.

Related Documentation

- Graceful Restart Concepts on page 105
- Graceful Restart System Requirements on page 106
- Graceful Restart on Logical Systems on page 112
- Verifying Graceful Restart Operation on page 128
- Example: Configuring Graceful Restart on page 130

Verifying Graceful Restart Operation

This topic contains the following sections:

- Graceful Restart Operational Mode Commands on page 128
- Verifying BGP Graceful Restart on page 129
- Verifying IS-IS and OSPF Graceful Restart on page 129
- Verifying CCC and TCC Graceful Restart on page 130

Graceful Restart Operational Mode Commands

To verify proper operation of graceful restart, use the following commands:

- **show bgp neighbor** (for BGP graceful restart)
- **show log** (for IS-IS and OSPF/OSPFv3 graceful restart)
- **show (ospf | ospfv3) overview** (for OSPF/OSPFv3 graceful restart)
- **show rsvp neighbor detail** (for RSVP graceful restart—helper router)
- **show rsvp version** (for RSVP graceful restart—restarting router)
- **show ldp session detail** (for LDP graceful restart)
- **show connections** (for CCC and TCC graceful restart)
- **show route instance detail** (for Layer 3 VPN graceful restart and for any protocols using graceful restart in a routing instance)
- **show route protocol l2vpn** (for Layer 2 VPN graceful restart)

For more information about these commands and a description of their output fields, see the Junos OS Operational Mode Commands.

Verifying BGP Graceful Restart

To view graceful restart information for BGP sessions, use the **show bgp neighbor** command:

```
user@PE1> show bgp neighbor 192.255.10.1
Peer: 192.255.10.1+179 AS 64595 Local: 192.255.5.1+1106 AS 64595
  Type: Internal    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ static ]
  Options:<Preference LocalAddress HoldTime GracefulRestart Damping PeerAS Refresh>

  Local Address: 192.255.5.1 Holdtime: 90 Preference: 170
  IPSec SA Name: hope
  Number of flaps: 0
  Peer ID: 192.255.10.1     Local ID: 192.255.5.1      Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 180
 Stale routes from peer are kept for: 180
 Restart time requested by this peer: 300
  NLRI that peer supports restart for: inet-unicast
  NLRI that peer saved forwarding for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Table inet.0 Bit: 10000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 0
    Received prefixes: 0
    Suppressed due to damping: 0
  Last traffic (seconds): Received 19   Sent 19   Checked 19
  Input messages:  Total 2      Updates 1       Refreshes 0     Octets 42
  Output messages: Total 3      Updates 0       Refreshes 0     Octets 116
  Output Queue[0]: 0
```

Verifying IS-IS and OSPF Graceful Restart

To view graceful restart information for IS-IS and OSPF, configure traceoptions (see "Tracking Graceful Restart Events" on page 120).

Here is the output of a traceoptions log from an OSPF restarting router:

```
Oct  8 05:20:12 Restart mode - sending grace lsas
Oct  8 05:20:12 Restart mode - estimated restart duration timer triggered
Oct  8 05:20:13 Restart mode - Sending more grace lsas
```

Here is the output of a traceoptions log from an OSPF helper router:

```
Oct  8 05:20:14 Helper mode for neighbor 192.255.5.1
Oct  8 05:20:14  Received multiple grace lsa from 192.255.5.1
```

Verifying CCC and TCC Graceful Restart

To view graceful restart information for CCC and TCC connections, use the **show connections** command. The following example assumes four remote interface CCC connections between CE1 and CE2:

```
user@PE1> show connections
CCC and TCC connections [Link Monitoring On]
  Legend for status (St)              Legend for connection types
  UN -- uninitialized                 if-sw:  interface switching
  NP -- not present                   rmt-if: remote interface switching
  WE -- wrong encapsulation           lsp-sw: LSP switching
  DS -- disabled
  Dn -- down                          Legend for circuit types
  -> -- only outbound conn is up      intf -- interface
  <- -- only inbound  conn is up      tlsp -- transmit LSP
  Up -- operational                   rlsp -- receive LSP
  RmtDn -- remote CCC down
  Restart -- restarting

CCC Graceful restart : Restarting

Connection/Circuit                  Type    St      Time last up     # Up trans
CE1-CE2-0                           rmt-if  Restart -----                     0
  fe-1/1/0.0                          intf  Up
  PE1-PE2-0                           tlsp  Up
  PE2-PE1-0                           rlsp  Up
CE1-CE2-1                           rmt-if  Restart -----                     0
  fe-1/1/0.1                          intf  Up
  PE1-PE2-1                           tlsp  Up
  PE2-PE1-1                           rlsp  Up
CE1-CE2-2                           rmt-if  Restart -----                     0
  fe-1/1/0.2                          intf  Up
  PE1-PE2-2                           tlsp  Up
  PE2-PE1-2                           rlsp  Up
CE1-CE2-3                           rmt-if  Restart -----                     0
  fe-1/1/0.3                          intf  Up
  PE1-PE2-3                           tlsp  Up
  PE2-PE1-3                           rlsp  Up
```

Related Documentation

- Graceful Restart Concepts on page 105
- Configuring Graceful Restart for QFabric Systems

Example: Configuring Graceful Restart

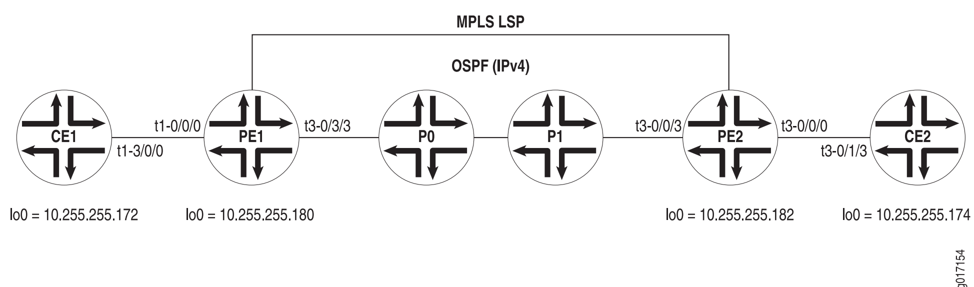
To enable graceful restart, include the **graceful-restart** statement at the **[edit routing-instance *instance-name* routing-options]** or **[edit routing-options]** hierarchy level as well as in the protocol level.

For example:

```
protocols {
  bgp {
    group ext {
      graceful-restart;
    }
  }
}
routing-options {
  graceful-restart;
}
```

Figure 7 on page 131 shows a standard MPLS VPN network. Routers CE1 and CE2 are customer edge routers, PE1 and PE2 are provider edge routers, and P0 is a provider core router. Several Layer 3 VPNs are configured across this network, as well as one Layer 2 VPN. Interfaces are shown in the diagram and are not included in the configuration example that follows.

Figure 7: Layer 3 VPN Graceful Restart Topology



Router CE1

On Router CE1, configure the following protocols on the logical interfaces of **t3-3/1/0**: OSPF on unit 101, RIP on unit 102, BGP on unit 103, and IS-IS on unit 512. Also configure graceful restart, BGP, IS-IS, OSPF, and RIP on the main instance to be able to connect to the routing instances on Router PE1.

```
[edit]
interfaces {
  t3-3/1/0 {
    encapsulation frame-relay;
    unit 100 {
      dlci 100;
      family inet {
        address 10.96.100.2/30;
      }
    }
    unit 101 {
      dlci 101;
      family inet {
        address 10.96.101.2/30;
      }
    }
    unit 102 {
      dlci 102;
      family inet {
        address 10.96.102.2/30;
```

```
            }
         }
         unit 103 {
            dlci 103;
            family inet {
               address 10.96.103.2/30;
            }
         }
         unit 512 {
            dlci 512;
            family inet {
               address 10.96.252.1/30;
            }
         }
      }
      lo0 {
         unit 0 {
            family inet {
               address 10.245.14.172/32;
               primary;
            }
            address 10.96.110.1/32;
            address 10.96.111.1/32;
            address 10.96.112.1/32;
            address 10.96.113.1/32;
            address 10.96.116.1/32;
         }
         family iso {
            address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4172.00;
         }
      }
   }
   routing-options {
      graceful-restart;
      autonomous-system 65100;
   }
   protocols {
      bgp {
         group CE-PE-INET {
            type external;
            export BGP_INET_LB_DIRECT;
            neighbor 10.96.103.1 {
               local-address 10.96.103.2;
               family inet {
                  unicast;
               }
               peer-as 65103;
            }
         }
      }
      isis {
         export ISIS_L2VPN_LB_DIRECT;
         interface t3-3/1/0.512;
      }
      ospf {
         export OSPF_LB_DIRECT;
```



```
        area 0.0.0.0 {
            interface t3-3/1/0.101;
        }
    }
    rip {
        group RIP {
            export RIP_LB_DIRECT;
            neighbor t3-3/1/0.102;
        }
    }
}
policy-options {
    policy-statement OSPF_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.101.0/30 exact;
                route-filter 10.96.111.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement RIP_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.102.0/30 exact;
                route-filter 10.96.112.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement BGP_INET_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.103.0/30 exact;
                route-filter 10.96.113.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement ISIS_L2VPN_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.116.1/32 exact;
```

```
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
}
```

Router PE1 On Router PE1, configure graceful restart in the master instance, along with BGP, OSPF, MPLS, and LDP. Next, configure several protocol-specific instances of graceful restart. By including instances for BGP, OSPF, Layer 2 VPNs, RIP, and static routes, you can observe the wide range of options available when you implement graceful restart. Configure the following protocols in individual instances on the logical interfaces of **t3-0/0/0**: a static route on unit 100, OSPF on unit 101, RIP on unit 102, BGP on unit 103, and Frame Relay on unit 512 for the Layer 2 VPN instance.

```
[edit]
interfaces {
    t3-0/0/0 {
        dce;
        encapsulation frame-relay-ccc;
        unit 100 {
            dlci 100;
            family inet {
                address 10.96.100.1/30;
            }
            family mpls;
        }
        unit 101 {
            dlci 101;
            family inet {
                address 10.96.101.1/30;
            }
            family mpls;
        }
        unit 102 {
            dlci 102;
            family inet {
                address 10.96.102.1/30;
            }
            family mpls;
        }
        unit 103 {
            dlci 103;
            family inet {
                address 10.96.103.1/30;
            }
            family mpls;
        }
        unit 512 {
            encapsulation frame-relay-ccc;
            dlci 512;
        }
    }
    t1-0/1/0 {
```



```
    unit 0 {
      family inet {
        address 10.96.0.2/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.245.14.176/32;
      }
      family iso {
        address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4176.00;
      }
    }
  }
}
routing-options {
  graceful-restart;
  router-id 10.245.14.176;
  autonomous-system 69;
}
protocols {
  mpls {
    interface all;
  }
  bgp {
    group PEPE {
      type internal;
      neighbor 10.245.14.182 {
        local-address 10.245.14.176;
        family inet-vpn {
          unicast;
        }
        family l2vpn {
          unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface t1-0/1/0.0;
      interface fxp0.0 {
        disable;
      }
      interface lo0.0 {
        passive;
      }
    }
  }
  ldp {
    interface all;
  }
}
```

```
policy-options {
    policy-statement STATIC-import {
        from community STATIC;
        then accept;
    }
    policy-statement STATIC-export {
        then {
            community add STATIC;
            accept;
        }
    }
    policy-statement OSPF-import {
        from community OSPF;
        then accept;
    }
    policy-statement OSPF-export {
        then {
            community add OSPF;
            accept;
        }
    }
    policy-statement RIP-import {
        from community RIP;
        then accept;
    }
    policy-statement RIP-export {
        then {
            community add RIP;
            accept;
        }
    }
    policy-statement BGP-INET-import {
        from community BGP-INET;
        then accept;
    }
    policy-statement BGP-INET-export {
        then {
            community add BGP-INET;
            accept;
        }
    }
    policy-statement L2VPN-import {
        from community L2VPN;
        then accept;
    }
    policy-statement L2VPN-export {
        then {
            community add L2VPN;
            accept;
        }
    }
    community BGP-INET members target:69:103;
    community L2VPN members target:69:512;
    community OSPF members target:69:101;
    community RIP members target:69:102;
    community STATIC members target:69:100;
```



```
}
routing-instances {
  BGP-INET {
    instance-type vrf;
    interface t3-0/0/0.103;
    route-distinguisher 10.245.14.176:103;
    vrf-import BGP-INET-import;
    vrf-export BGP-INET-export;
    routing-options {
      graceful-restart;
      autonomous-system 65103;
    }
    protocols {
      bgp {
        group BGP-INET {
          type external;
          export BGP-INET-import;
          neighbor 10.96.103.2 {
            local-address 10.96.103.1;
            family inet {
              unicast;
            }
            peer-as 65100;
          }
        }
      }
    }
  }
  L2VPN {
    instance-type l2vpn;
    interface t3-0/0/0.512;
    route-distinguisher 10.245.14.176:512;
    vrf-import L2VPN-import;
    vrf-export L2VPN-export;
    protocols {# There is no graceful-restart statement for Layer 2 VPN instances.
      l2vpn {
        encapsulation-type frame-relay;
        site CE1-ISIS {
          site-identifier 512;
          interface t3-0/0/0.512 {
            remote-site-id 612;
          }
        }
      }
    }
  }
  OSPF {
    instance-type vrf;
    interface t3-0/0/0.101;
    route-distinguisher 10.245.14.176:101;
    vrf-import OSPF-import;
    vrf-export OSPF-export;
    routing-options {
      graceful-restart;
    }
    protocols {
```



```
            ospf {
                export OSPF-import;
                area 0.0.0.0 {
                    interface all;
                }
            }
        }
    }
    RIP {
        instance-type vrf;
        interface t3-0/0/0.102;
        route-distinguisher 10.245.14.176:102;
        vrf-import RIP-import;
        vrf-export RIP-export;
        routing-options {
            graceful-restart;
        }
        protocols {
            rip {
                group RIP {
                    export RIP-import;
                    neighbor t3-0/0/0.102;
                }
            }
        }
    }
    STATIC {
        instance-type vrf;
        interface t3-0/0/0.100;
        route-distinguisher 10.245.14.176:100;
        vrf-import STATIC-import;
        vrf-export STATIC-export;
        routing-options {
            graceful-restart;
            static {
                route 10.96.110.1/32 next-hop t3-0/0/0.100;
            }
        }
    }
}
```

Router P0 On Router P0, configure graceful restart in the main instance, along with OSPF, MPLS, and LDP. This allows the protocols on the PE routers to reach one another.

```
[edit]
interfaces {
    t3-0/1/3 {
        unit 0 {
            family inet {
                address 10.96.0.5/30;
            }
            family mpls;
        }
    }
    t1-0/2/0 {
        unit 0 {
```



```
                family inet {
                    address 10.96.0.1/30;
                }
                family mpls;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.245.14.174/32;
                }
                family iso {
                    address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4174.00;
                }
            }
        }
    }
    routing-options {
        graceful-restart;
        router-id 10.245.14.174;
        autonomous-system 69;
    }
    protocols {
        mpls {
            interface all;
        }
        ospf {
            area 0.0.0.0 {
                interface t1-0/2/0.0;
                interface t3-0/1/3.0;
                interface fxp0.0 {
                    disable;
                }
                interface lo0.0 {
                    passive;
                }
            }
        }
        ldp {
            interface all;
        }
    }
```

Router PE2

On Router PE2, configure BGP, OSPF, MPLS, LDP, and graceful restart in the master instance. Configure the following protocols in individual instances on the logical interfaces of **t1-0/1/3**: a static route on unit 200, OSPF on unit 201, RIP on unit 202, BGP on unit 203, and Frame Relay on unit 612 for the Layer 2 VPN instance. Also configure protocol-specific graceful restart in all routing instances, except the Layer 2 VPN instance.

```
[edit]
interfaces {
    t3-0/0/0 {
        unit 0 {
            family inet {
                address 10.96.0.6/30;
            }
```

```
                family mpls;
            }
        }
        t1-0/1/3 {
            dce;
            encapsulation frame-relay-ccc;
            unit 200 {
                dlci 200;
                family inet {
                    address 10.96.200.1/30;
                }
                family mpls;
            }
            unit 201 {
                dlci 201;
                family inet {
                    address 10.96.201.1/30;
                }
                family mpls;
            }
            unit 202 {
                dlci 202;
                family inet {
                    address 10.96.202.1/30;
                }
                family mpls;
            }
            unit 203 {
                dlci 203;
                family inet {
                    address 10.96.203.1/30;
                }
                family mpls;
            }
            unit 612 {
                encapsulation frame-relay-ccc;
                dlci 612;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.245.14.182/32;
                }
                family iso {
                    address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4182.00;
                }
            }
        }
    }
    routing-options {
        graceful-restart;
        router-id 10.245.14.182;
        autonomous-system 69;
    }
    protocols {
```



```
mpls {
  interface all;
}
bgp {
  group PEPE {
    type internal;
    neighbor 10.245.14.176 {
      local-address 10.245.14.182;
      family inet-vpn {
        unicast;
      }
      family l2vpn {
        unicast;
      }
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface t3-0/0/0.0;
    interface fxp0.0 {
      disable;
    }
    interface lo0.0 {
      passive;
    }
  }
}
ldp {
  interface all;
}
policy-options {
  policy-statement STATIC-import {
    from community STATIC;
    then accept;
  }
  policy-statement STATIC-export {
    then {
      community add STATIC;
      accept;
    }
  }
  policy-statement OSPF-import {
    from community OSPF;
    then accept;
  }
  policy-statement OSPF-export {
    then {
      community add OSPF;
      accept;
    }
  }
  policy-statement RIP-import {
    from community RIP;
    then accept;
  }
```

```
    policy-statement RIP-export {
        then {
            community add RIP;
            accept;
        }
    }
    policy-statement BGP-INET-import {
        from community BGP-INET;
        then accept;
    }
    policy-statement BGP-INET-export {
        then {
            community add BGP-INET;
            accept;
        }
    }
    policy-statement L2VPN-import {
        from community L2VPN;
        then accept;
    }
    policy-statement L2VPN-export {
        then {
            community add L2VPN;
            accept;
        }
    }
    community BGP-INET members target:69:103;
    community L2VPN members target:69:512;
    community OSPF members target:69:101;
    community RIP members target:69:102;
    community STATIC members target:69:100;
}
routing-instances {
    BGP-INET {
        instance-type vrf;
        interface t1-0/1/3.203;
        route-distinguisher 10.245.14.182:203;
        vrf-import BGP-INET-import;
        vrf-export BGP-INET-export;
        routing-options {
            graceful-restart;
            autonomous-system 65203;
        }
        protocols {
            bgp {
                group BGP-INET {
                    type external;
                    export BGP-INET-import;
                    neighbor 10.96.203.2 {
                        local-address 10.96.203.1;
                        family inet {
                            unicast;
                        }
                        peer-as 65200;
                    }
                }
```



```
    }
  }
}
L2VPN {
  instance-type l2vpn;
  interface t1-0/1/3.612;
  route-distinguisher 10.245.14.182:612;
  vrf-import L2VPN-import;
  vrf-export L2VPN-export;
  protocols {# There is no graceful-restart statement for Layer 2 VPN instances.
    l2vpn {
      encapsulation-type frame-relay;
      site CE2-ISIS {
        site-identifier 612;
        interface t1-0/1/3.612 {
          remote-site-id 512;
        }
      }
    }
  }
}
OSPF {
  instance-type vrf;
  interface t1-0/1/3.201;
  route-distinguisher 10.245.14.182:201;
  vrf-import OSPF-import;
  vrf-export OSPF-export;
  routing-options {
    graceful-restart;
  }
  protocols {
    ospf {
      export OSPF-import;
      area 0.0.0.0 {
        interface all;
      }
    }
  }
}
RIP {
  instance-type vrf;
  interface t1-0/1/3.202;
  route-distinguisher 10.245.14.182:202;
  vrf-import RIP-import;
  vrf-export RIP-export;
  routing-options {
    graceful-restart;
  }
  protocols {
    rip {
      group RIP {
        export RIP-import;
        neighbor t1-0/1/3.202;
      }
    }
  }
```

```
        }
        STATIC {
            instance-type vrf;
            interface t1-0/1/3.200;
            route-distinguisher 10.245.14.182:200;
            vrf-import STATIC-import;
            vrf-export STATIC-export;
            routing-options {
                graceful-restart;
                static {
                    route 10.96.210.1/32 next-hop t1-0/1/3.200;
                }
            }
        }
    }
}
```

Router CE2

On Router CE2, complete the Layer 2 and Layer 3 VPN configuration by mirroring the protocols already set on Routers PE2 and CE1. Specifically, configure the following on the logical interfaces of **t1-0/0/3**: OSPF on unit 201, RIP on unit 202, BGP on unit 203, and IS-IS on unit 612. Finally, configure graceful restart, BGP, IS-IS, OSPF, and RIP on the main instance to be able to connect to the routing instances on Router PE2.

```
[edit]
interfaces {
    t1-0/0/3 {
        encapsulation frame-relay;
        unit 200 {
            dlci 200;
            family inet {
                address 10.96.200.2/30;
            }
        }
        unit 201 {
            dlci 201;
            family inet {
                address 10.96.201.2/30;
            }
        }
        unit 202 {
            dlci 202;
            family inet {
                address 10.96.202.2/30;
            }
        }
        unit 203 {
            dlci 203;
            family inet {
                address 10.96.203.2/30;
            }
        }
        unit 512 {
            dlci 512;
            family inet {
                address 10.96.252.2/30;
            }
```

```
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.180/32 {
                    primary;
                }
                address 10.96.210.1/32;
                address 10.96.111.1/32;
                address 10.96.212.1/32;
                address 10.96.213.1/32;
                address 10.96.216.1/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4180.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    autonomous-system 65200;
}
protocols {
    bgp {
        group CE-PE-INET {
            type external;
            export BGP_INET_LB_DIRECT;
            neighbor 10.96.203.1 {
                local-address 10.96.203.2;
                family inet {
                    unicast;
                }
                peer-as 65203;
            }
        }
    }
    isis {
        export ISIS_L2VPN_LB_DIRECT;
        interface t1-0/0/3.612;
    }
    ospf {
        export OSPF_LB_DIRECT;
        area 0.0.0.0 {
            interface t1-0/0/3.201;
        }
    }
    rip {
        group RIP {
            export RIP_LB_DIRECT;
            neighbor t1-0/0/3.202;
        }
    }
}
policy-options {
```



```
    policy-statement OSPF_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.201.0/30 exact;
                route-filter 10.96.211.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement RIP_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.202.0/30 exact;
                route-filter 10.96.212.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement BGP_INET_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.203.0/30 exact;
                route-filter 10.96.213.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement ISIS_L2VPN_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.216.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
}
```


Router PE1 Status Before a Restart

The following example displays neighbor relationships on Router PE1 before a restart happens:

```
user@PE1> show bgp neighbor
Peer: 10.96.103.2+3785 AS 65100 Local: 10.96.103.1+179 AS 65103
  Type: External    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ BGP-INET-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS
 Refresh>
  Address families configured: inet-unicast
  Local Address: 10.96.103.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.96.110.1      Local ID: 10.96.103.1      Active Holdtime: 90
  Keepalive Interval: 30
  Local Interface: t3-0/0/0.103
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI peer can save forwarding state: inet-unicast
  NLRI that peer saved forwarding for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Table BGP-INET.inet.0 Bit: 30001
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:           0
    Received prefixes:         0
    Suppressed due to damping:  0
  Last traffic (seconds): Received 8    Sent 3    Checked 3
  Input messages:  Total 15     Updates 0      Refreshes 0     Octets 321
  Output messages: Total 18     Updates 2      Refreshes 0     Octets 450
  Output Queue[2]: 0

Peer: 10.245.14.182+4701 AS 69    Local: 10.245.14.176+179 AS 69
  Type: Internal    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily
Rib-group Refresh>
  Address families configured: inet-vpn-unicast l2vpn
  Local Address: 10.245.14.176 Holdtime: 90 Preference: 170
  Number of flaps: 1
  Peer ID: 10.245.14.182    Local ID: 10.245.14.176    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast l2vpn
  NLRI advertised by peer: inet-vpn-unicast l2vpn
  NLRI for this session: inet-vpn-unicast l2vpn
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-vpn-unicast l2vpn
  NLRI peer can save forwarding state: inet-vpn-unicast l2vpn
```



```
NLRI that peer saved forwarding for: inet-vpn-unicast l2vpn
NLRI that restart is negotiated for: inet-vpn-unicast l2vpn
NLRI of all end-of-rib markers sent: inet-vpn-unicast l2vpn
Table bgp.l3vpn.0 Bit: 10000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:           0
  Received prefixes:         0
  Suppressed due to damping:  0
Table bgp.l2vpn.0 Bit: 20000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:           1
  Received prefixes:         1
  Suppressed due to damping:  0
Table BGP-INET.inet.0 Bit: 30000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:           0
  Received prefixes:         0
  Suppressed due to damping:  0
Table OSPF.inet.0 Bit: 60000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:           0
  Received prefixes:         0
  Suppressed due to damping:  0
Table RIP.inet.0 Bit: 70000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:           0
  Received prefixes:         0
  Suppressed due to damping:  0
Table STATIC.inet.0 Bit: 80000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:           0
  Received prefixes:         0
  Suppressed due to damping:  0
Table L2VPN.l2vpn.0 Bit: 90000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:           1
  Received prefixes:         1
  Suppressed due to damping:  0
Last traffic (seconds): Received 28   Sent 28   Checked 28
Input messages:  Total 2      Updates 0      Refreshes 0     Octets 86
Output messages: Total 13     Updates 10     Refreshes 0     Octets 1073
Output Queue[0]: 0
Output Queue[1]: 0
Output Queue[2]: 0
Output Queue[3]: 0
Output Queue[4]: 0
Output Queue[5]: 0
```



```
  Output Queue[6]: 0
  Output Queue[7]: 0
  Output Queue[8]: 0

user@PE1> show route instance detail
master:
  Router ID: 10.245.14.176
  Type: forwarding        State: Active
  Restart State: Complete Path selection timeout: 300
  Tables:
    inet.0                 : 17 routes (15 active, 0 holddown, 1 hidden)
    Restart Complete
    inet.3                 : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    iso.0                  : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
    mpls.0                 : 19 routes (19 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l3vpn.0            : 10 routes (10 active, 0 holddown, 0 hidden)
    Restart Complete
    inet6.0                : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l2vpn.0            : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
BGP-INET:
  Router ID: 10.96.103.1
  Type: vrf               State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.103
  Route-distinguisher: 10.245.14.176:103
  Vrf-import: [ BGP-INET-import ]
  Vrf-export: [ BGP-INET-export ]
  Tables:
    BGP-INET.inet.0        : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Complete
L2VPN:
  Router ID: 0.0.0.0
  Type: l2vpn             State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.512
  Route-distinguisher: 10.245.14.176:512
  Vrf-import: [ L2VPN-import ]
  Vrf-export: [ L2VPN-export ]
  Tables:
    L2VPN.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
OSPF:
  Router ID: 10.96.101.1
  Type: vrf               State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.101
  Route-distinguisher: 10.245.14.176:101
  Vrf-import: [ OSPF-import ]
  Vrf-export: [ OSPF-export ]
  Tables:
    OSPF.inet.0            : 8 routes (7 active, 0 holddown, 0 hidden)
    Restart Complete
RIP:
```

```
  Router ID: 10.96.102.1
  Type: vrf              State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.102
  Route-distinguisher: 10.245.14.176:102
  Vrf-import: [ RIP-import ]
  Vrf-export: [ RIP-export ]
  Tables:
    RIP.inet.0             : 6 routes (6 active, 0 holddown, 0 hidden)
    Restart Complete
STATIC:
  Router ID: 10.96.100.1
  Type: vrf              State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.100
  Route-distinguisher: 10.245.14.176:100
  Vrf-import: [ STATIC-import ]
  Vrf-export: [ STATIC-export ]
  Tables:
    STATIC.inet.0          : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Complete
__juniper_private1__:
  Router ID: 0.0.0.0
  Type: forwarding       State: Active

user@PE1> show route protocol l2vpn
inet.0: 16 destinations, 17 routes (15 active, 0 holddown, 1 hidden)
Restart Complete
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
BGP-INET.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
Restart Complete
OSPF.inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
Restart Complete
RIP.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
Restart Complete
STATIC.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete
mpls.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
800003            *[L2VPN/7] 00:06:00
                   > via t3-0/0/0.512, Pop       Offset: 4
t3-0/0/0.512      *[L2VPN/7] 00:06:00
                   > via t1-0/1/0.0, Push 800003, Push 100004(top) Offset: -4
bgp.l3vpn.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Restart Complete
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
L2VPN.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
10.245.14.176:512:512:611/96
                  *[L2VPN/7] 00:06:01
                     Discard
```



```
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete
```

Router PE1 Status During a Restart

Before you can verify that graceful restart is working, you must simulate a router restart. To cause the routing process to refresh and simulate a restart, use the **restart routing** operational mode command:

```
user@PE1> restart routing
Routing protocol daemon started, pid 3558
```

The following sample output is captured during the router restart:

```
user@PE1> show bgp neighbor
Peer: 10.96.103.2     AS 65100 Local: 10.96.103.1     AS 65103
  Type: External    State: Active        Flags: <ImportEval>
  Last State: Idle          Last Event: Start
  Last Error: None
  Export: [ BGP-INET-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS
 Refresh>
  Address families configured: inet-unicast
  Local Address: 10.96.103.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
Peer: 10.245.14.182+179 AS 69    Local: 10.245.14.176+2131 AS 69
  Type: Internal    State: Established    Flags: <ImportEval>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily
Rib-group Refresh>
  Address families configured: inet-vpn-unicast l2vpn
  Local Address: 10.245.14.176 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.245.14.182    Local ID: 10.245.14.176    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast l2vpn
  NLRI advertised by peer: inet-vpn-unicast l2vpn
  NLRI for this session: inet-vpn-unicast l2vpn
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-vpn-unicast l2vpn
  NLRI peer can save forwarding state: inet-vpn-unicast l2vpn
  NLRI that peer saved forwarding for: inet-vpn-unicast l2vpn
  NLRI that restart is negotiated for: inet-vpn-unicast l2vpn
  NLRI of received end-of-rib markers: inet-vpn-unicast l2vpn
  Table bgp.l3vpn.0 Bit: 10000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:           10
    Received prefixes:         10
    Suppressed due to damping:  0
  Table bgp.l2vpn.0 Bit: 20000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:           1
    Received prefixes:         1
    Suppressed due to damping:  0
  Table BGP-INET.inet.0 Bit: 30000
```



```
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:           2
    Received prefixes:         2
    Suppressed due to damping:  0
Table OSPF.inet.0 Bit: 60000
    RIB State: BGP restart is complete
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:           2
    Received prefixes:         2
    Suppressed due to damping:  0
  Table RIP.inet.0 Bit: 70000
    RIB State: BGP restart is complete
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:           2
    Received prefixes:         2
    Suppressed due to damping:  0
  Table STATIC.inet.0 Bit: 80000
    RIB State: BGP restart is complete
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:           1
    Received prefixes:         1
    Suppressed due to damping:  0
  Table L2VPN.l2vpn.0 Bit: 90000
    RIB State: BGP restart is complete
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:           1
    Received prefixes:         1
    Suppressed due to damping:  0
  Last traffic (seconds): Received 0    Sent 0    Checked 0
  Input messages:  Total 14     Updates 13      Refreshes 0     Octets 1053
  Output messages: Total 3      Updates 0       Refreshes 0     Octets 105
  Output Queue[0]: 0
  Output Queue[1]: 0
  Output Queue[2]: 0
  Output Queue[3]: 0
  Output Queue[4]: 0
  Output Queue[5]: 0
  Output Queue[6]: 0
  Output Queue[7]: 0
  Output Queue[8]: 0

user@PE1> show route instance detail
master:
  Router ID: 10.245.14.176
  Type: forwarding        State: Active
  Restart State: Pending  Path selection timeout: 300
  Tables:
    inet.0                 : 17 routes (15 active, 1 holddown, 1 hidden)
    Restart Pending: OSPF LDP
    inet.3                 : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Pending: OSPF LDP
    iso.0                  : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
    mpls.0                 : 23 routes (23 active, 0 holddown, 0 hidden)
    Restart Pending: LDP VPN
```



```
    bgp.l3vpn.0           : 10 routes (10 active, 0 holddown, 0 hidden)
    Restart Pending: BGP VPN
    inet6.0               : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l2vpn.0           : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Pending: BGP VPN
BGP-INET:
  Router ID: 10.96.103.1
  Type: vrf              State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.103
  Route-distinguisher: 10.245.14.176:103
  Vrf-import: [ BGP-INET-import ]
  Vrf-export: [ BGP-INET-export ]
  Tables:
    BGP-INET.inet.0       : 6 routes (5 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
L2VPN:
  Router ID: 0.0.0.0
  Type: l2vpn            State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.512
  Route-distinguisher: 10.245.14.176:512
  Vrf-import: [ L2VPN-import ]
  Vrf-export: [ L2VPN-export ]
  Tables:
    L2VPN.l2vpn.0         : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Pending: VPN L2VPN
OSPF:
  Router ID: 10.96.101.1
  Type: vrf              State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.101
  Route-distinguisher: 10.245.14.176:101
  Vrf-import: [ OSPF-import ]
  Vrf-export: [ OSPF-export ]
  Tables:
    OSPF.inet.0           : 8 routes (7 active, 1 holddown, 0 hidden)
    Restart Pending: OSPF VPN
RIP:
  Router ID: 10.96.102.1
  Type: vrf              State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.102
  Route-distinguisher: 10.245.14.176:102
  Vrf-import: [ RIP-import ]
  Vrf-export: [ RIP-export ]
  Tables:
    RIP.inet.0            : 8 routes (6 active, 2 holddown, 0 hidden)
    Restart Pending: RIP VPN
STATIC:
  Router ID: 10.96.100.1
  Type: vrf              State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.100
  Route-distinguisher: 10.245.14.176:100
```

```
  Vrf-import: [ STATIC-import ]
  Vrf-export: [ STATIC-export ]
  Tables:
    STATIC.inet.0          : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
__juniper_private1__:
  Router ID: 0.0.0.0
  Type: forwarding        State: Active
```

user@PE1> **show route instance summary**

```
Instance           Type           Primary rib           Active/holddown/hidden
master             forwarding
                                  inet.0                15/0/1
                                  iso.0                 1/0/0
                                  mpls.0                35/0/0
                                  l3vpn.0               0/0/0
                                  inet6.0               2/0/0
                                  l2vpn.0               0/0/0
                                  l2circuit.0           0/0/0
BGP-INET           vrf
                                  BGP-INET.inet.0       5/0/0
                                  BGP-INET.iso.0        0/0/0
                                  BGP-INET.inet6.0      0/0/0
L2VPN              l2vpn
                                  L2VPN.inet.0          0/0/0
                                  L2VPN.iso.0           0/0/0
                                  L2VPN.inet6.0         0/0/0
                                  L2VPN.l2vpn.0         2/0/0
OSPF               vrf
                                  OSPF.inet.0           7/0/0
                                  OSPF.iso.0            0/0/0
                                  OSPF.inet6.0          0/0/0
RIP                vrf
                                  RIP.inet.0            6/0/0
                                  RIP.iso.0             0/0/0
                                  RIP.inet6.0           0/0/0
STATIC             vrf
                                  STATIC.inet.0         4/0/0
                                  STATIC.iso.0          0/0/0
                                  STATIC.inet6.0        0/0/0
__juniper_private1__ forwarding
                                  __juniper_priva.inet.0 0/0/0
                                  __juniper_privat.iso.0 0/0/0
                                  __juniper_priv.inet6.0 0/0/0
```

user@PE1> **show route protocol l2vpn**

```
inet.0: 16 destinations, 17 routes (15 active, 1 holddown, 1 hidden)
Restart Pending: OSPF LDP

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Pending: OSPF LDP

BGP-INET.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
Restart Pending: VPN

OSPF.inet.0: 7 destinations, 8 routes (7 active, 1 holddown, 0 hidden)
Restart Pending: OSPF VPN

RIP.inet.0: 6 destinations, 8 routes (6 active, 2 holddown, 0 hidden)
Restart Pending: RIP VPN
```



```
STATIC.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Pending: VPN

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

mpls.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
Restart Pending: LDP VPN
+ = Active Route, - = Last Active, * = Both

800001             *[L2VPN/7] 00:00:13
                    > via t3-0/0/0.512, Pop       Offset: 4
t3-0/0/0.512       *[L2VPN/7] 00:00:13
                    > via t1-0/1/0.0, Push 800003, Push 100004(top) Offset: -4

bgp.l3vpn.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Restart Pending: BGP VPN

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

L2VPN.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Pending: VPN L2VPN
+ = Active Route, - = Last Active, * = Both

10.245.14.176:512:512:611/96
                   *[L2VPN/7] 00:00:13
                      Discard
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Pending: BGP VPN
```

Related Documentation

- Enabling Graceful Restart on page 113
- Configuring Routing Protocols Graceful Restart on page 114
- Configuring Graceful Restart for MPLS-Related Protocols on page 123
- Configuring VPN Graceful Restart on page 125
- Configuring Logical System Graceful Restart on page 127
- Verifying Graceful Restart Operation on page 128

CHAPTER 16

Summary of Graceful Restart Configuration Statements

This chapter provides a reference for each of the graceful restart configuration statements. The statements are organized alphabetically.

disable

Syntax disable;

Hierarchy Level [edit logical-systems *logical-system-name* protocols (bgp | isis | ldp | ospf | ospf3 | pim | rip | ripng | rsvp) graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols (bgp | ldp | ospf | ospf3 | pim) graceful-restart],
[edit protocols (bgp | esis | isis | ospf | ospf3 | ldp | pim | rip | ripng | rsvp) graceful-restart],
[edit protocols bgp group *group-name* graceful-restart],
[edit protocols bgp group *group-name* neighbor *ip-address* graceful-restart],
[edit routing-instances *routing-instance-name* protocols (bgp | ldp | ospf | ospf3 | pim) graceful-restart],
[edit routing-instances *routing-instance-name* routing-options graceful-restart],
[edit routing-options graceful-restart]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description Disable graceful restart.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling Graceful Restart on page 113
- Configuring Routing Protocols Graceful Restart on page 114
- Configuring Graceful Restart for MPLS-Related Protocols on page 123
- Configuring VPN Graceful Restart on page 125
- Configuring Logical System Graceful Restart on page 127

- Graceful Restart Configuration Statements
- Configuring Graceful Restart for QFabric Systems

graceful-restart (Enabling Globally)

Syntax

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

Hierchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure graceful restart globally to enable the feature. You cannot enable graceful restart for specific protocols unless graceful restart is also enabled globally.

For VPNs, the **graceful-restart** statement allows a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers.

For BGP, if you configure graceful restart after a BGP session has been established, the BGP session restarts and the peers negotiate graceful restart capabilities.

Default

Graceful restart is disabled by default.

Options

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Enabling Graceful Restart on page 113
- Configuring Routing Protocols Graceful Restart on page 114
- Configuring Graceful Restart for MPLS-Related Protocols on page 123
- Configuring VPN Graceful Restart on page 125
- Configuring Logical System Graceful Restart on page 127

- Graceful Restart Configuration Statements
- Configuring Graceful Restart for QFabric Systems

helper-disable (Multiple Protocols)

Syntax helper-disable;

Hierarchy Level [edit logical-systems *logical-system-name* protocols (isis | ldp | ospf | ospf3 | rsvp) graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols (ldp | ospf | ospf3) graceful-restart],
[edit protocols (isis | ldp | ospf | ospf3 | rsvp) graceful-restart],
[edit routing-instances *routing-instance-name* protocols (ldp | ospf | ospf3) graceful-restart]

Release Information Statement introduced before Junos OS Release 7.4.

Description Disable helper mode for graceful restart. When helper mode is disabled, a router cannot help a neighboring router that is attempting to restart.

Default Helper mode is enabled by default for these supported protocols: IS-IS, LDP, OSPF/OSPFv3, and RSVP.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Routing Protocols Graceful Restart on page 114
- Configuring Graceful Restart for MPLS-Related Protocols on page 123

helper-disable (OSPF)

Syntax helper-disable < both | restart-signaling | standard >;

Hierarchy Level [edit logical-systems *logical-system-name* protocols ospf graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ospf graceful-restart],
[edit protocols ospf graceful-restart],
[edit routing-instances *routing-instance-name* protocols ospf graceful-restart]

Release Information Statement introduced before Junos OS Release 7.4.
Options **both**, **restart-signaling**, and **standard** introduced in Junos OS Release 11.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Disable helper mode for graceful restart. When helper mode is disabled, a router cannot help a neighboring router that is attempting to restart. The last committed statement takes precedence over the previously configured statement.

Default Helper mode is enabled by default for OSPF.

Options **both**—(Optional) Disable helper mode for both standard and restart signaling-based graceful restart.

restart-signaling—(Optional) Disable helper mode for restart signaling-based graceful restart (based on RFC 4811, RFC 4812, and RFC 4813).

NOTE: Restart signaling-based helper mode is not supported for OSPFv3 configurations.

standard—(Optional) Disable helper mode for standard graceful restart (based on RFC 3623).

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Routing Protocols Graceful Restart on page 114
- Configuring Graceful Restart for MPLS-Related Protocols on page 123
- Configuring Graceful Restart for QFabric Systems

maximum-helper-recovery-time

Syntax maximum-helper-recovery-time *seconds*;

Hierarchy Level [edit protocols rsvp graceful-restart],
[edit logical-systems *logical-system-name* protocols rsvp graceful-restart]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the length of time the router retains the state of its Resource Reservation Protocol (RSVP) neighbors while they undergo a graceful restart.

Options *seconds*—Legnth of time that the router retains the state of its Resource Reservation Protocol (RSVP) neighbors while they undergo a graceful restart.

Range: 1 through 3600

Default: 180

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for RSVP, CCC, and TCC on page 124
- maximum-helper-restart-time (RSVP) on page 162

maximum-helper-restart-time (RSVP)

Syntax maximum-helper-restart-time *seconds*;

Hierarchy Level [edit protocols rsvp graceful-restart],
[edit logical-systems *logical-system-name* protocols rsvp graceful-restart]

Release Information Statement introduced in Junos OS Release 8.3.

Description Specify the length of time the router waits after it discovers that a neighboring router has gone down before it declares the neighbor down. This value is applied to all RSVP neighbor routers and should be based on the time that the slowest RSVP neighbor requires for restart.

Options *seconds*—The time the router waits after it discovers that a neighboring router has gone down before it declares the neighbor down.

Range: 1 through 1800

Default: 60

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for RSVP, CCC, and TCC on page 124
- maximum-helper-recovery-time on page 162

maximum-neighbor-reconnect-time

Syntax maximum-neighbor-reconnect-time *seconds*;

Hierarchy Level [edit protocols ldp graceful-restart],
[edit logical-systems *logical-system-name* protocols ldp graceful-restart],
[edit routing-instances *routing-instance-name* protocols ldp graceful-restart]

Release Information Statement introduced in Junos OS Release 9.1.

Description Specify the maximum length of time allowed to reestablish connection from a restarting neighbor.

Options ***seconds***—Maximum time allowed for reconnection.

Range: 30 through 300

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for LDP on page 125

maximum-neighbor-recovery-time (OBSOLETE)

Syntax maximum-neighbor-recovery-time *seconds*;

Hierarchy Level [edit protocols ldp graceful-restart],
[edit logical-systems *logical-system-name* protocols ldp graceful-restart],
[edit routing-instances *instance-name* protocols ldp graceful-restart]

Release Information Statement introduced in Junos OS Release 8.3.
Statement name changed from **maximum-recovery-time** to **maximum-neighbor-recovery-time** in Junos OS Release 9.1.

Description Specify the length of time the router retains the state of its Label Distribution Protocol (LDP) neighbors while they undergo a graceful restart.

Options ***seconds***—Time, in seconds, the router retains the state of its LDP neighbors while they undergo a graceful restart.

Range: 140 through 1900

Default: 240

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for LDP on page 125
- **no-strict-lsa-checking on page 164**
- **recovery-time on page 167**

no-strict-lsa-checking

Syntax no-strict-lsa-checking;

Hierarchy Level [edit protocols (ospf | ospf3) graceful-restart]

Release Information Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Disable strict OSPF link-state advertisement (LSA) checking to prevent the termination of graceful restart by a helping router.

Default By default, LSA checking is enabled.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for OSPF and OSPFv3 on page 117
- Configuring Graceful Restart for QFabric Systems
- maximum-neighbor-recovery-time
- recovery-time on page 167

notify-duration

Syntax notify-duration *seconds*;

Hierarchy Level [edit protocols (ospf | ospf3) graceful-restart],
[edit logical-systems *logical-system-name* protocols (ospf | ospf3) graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols (ospf | ospf3) graceful-restart],
[edit routing-instances *instance-name* protocols (ospf | ospf3) graceful-restart]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Specify the length of time the router notifies helper OSPF routers that it has completed graceful restart.

Options *seconds*—Length of time in the router notifies helper OSPF routers that it has completed graceful restart.

Range: 1 through 3600

Default: 30

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for OSPF and OSPFv3 on page 117
- Configuring Graceful Restart for QFabric Systems
- restart-duration on page 168

reconnect-time

Syntax reconnect-time *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols ldp graceful-restart],
[edit protocols ldp graceful-restart],
[edit routing-instances *routing-instance-name* protocols ldp graceful-restart]

Release Information Statement introduced in Junos OS Release 9.1.

Description Specify the length of time required to reestablish a Label Distribution Protocol (LDP) session after graceful restart.

Options ***seconds***—Time required for reconnection.

Range: 30 through 300

Default: 60 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring LDP Graceful Restart on LDP Configuration Guide
- Configuring Graceful Restart Options for LDP on page 125

recovery-time

Syntax recovery-time *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols ldp graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ldp graceful-restart],
[edit protocols ldp graceful-restart],
[edit routing-instances *routing-instance-name* protocols ldp graceful-restart]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the length of time a router waits for Label Distribution Protocol (LDP) neighbors to assist it with a graceful restart.

Options *seconds*—Time the router waits for LDP to restart gracefully.

Range: 120 through 1800

Default: 160

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for LDP on page 125
- maximum-neighbor-recovery-time
- no-strict-lsa-checking on page 164

restart-duration

Syntax restart-duration *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols (isis | ospf | ospf3 | pim) graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols (ospf | ospf3 | pim) graceful-restart],
[edit protocols (esis | isis | ospf | ospf3 | pim) graceful-restart],
[edit routing-instances *routing-instance-name* protocols (ospf | ospf3 | pim) graceful-restart],
[edit routing-options graceful-restart]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure the grace period for graceful restart globally.

Additionally, you can individually configure the duration of the graceful restart period for the End System-to-Intermediate System (ES-IS), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), and OSPFv3 protocols and for Protocol Independent Multicast (PIM) sparse mode.

Options ***seconds***—Time for the graceful restart period.

Range:

The range of values varies according to whether the graceful restart period is being set globally or for a particular protocol:

- **[edit routing-options graceful-restart]** (global setting)—120 through 900
- ES-IS—30 through 300
- IS-IS—30 through 300
- OSPF/OSPFv3—1 through 3600
- PIM—30 through 300

Default:

The default value varies according to whether the graceful restart period is being set globally or for a particular protocol:

- **[edit routing-options graceful-restart]** (global setting)—300
- ES-IS—180
- IS-IS—210
- OSPF/OSPFv3—180
- PIM—60

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling Graceful Restart on page 113
- Configuring Routing Protocols Graceful Restart on page 114
- Configuring Graceful Restart for MPLS-Related Protocols on page 123
- Configuring VPN Graceful Restart on page 125
- Configuring Graceful Restart for VPNs
- Configuring Logical System Graceful Restart on page 127
- Graceful Restart Configuration Statements
- Configuring Graceful Restart for QFabric Systems

restart-time (BGP Graceful Restart)

Syntax restart-time *seconds*;

Hierarchy Level [edit protocols (bgp | rip | ripng) graceful-restart],
[edit logical-systems *logical-system-name* protocols (bgp | rip | ripng) graceful-restart (Enabling Globally)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp graceful-restart],
[edit routing-instances *routing-instance-name* protocols bgp graceful-restart]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure the duration of the BGP, RIP, or next-generation RIP (RIPng) graceful restart period.

Options *seconds*—Length of time for the graceful restart period.

Range: 1 through 600 seconds

Default: Varies by protocol:

- BGP—120 seconds
- RIP and RIPng—60 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for BGP on page 115
- Configuring Graceful Restart Options for RIP and RIPng on page 118
- Configuring Graceful Restart for QFabric Systems
- stale-routes-time on page 170

stale-routes-time

Syntax stale-routes-time *seconds*;

Hierarchy Level [edit logical-systems *logical-routing-name* protocols bgp graceful-restart],
[edit logical-systems *logical-routing-name* routing-instances *routing-instance-name* protocols bgp graceful-restart],
[edit protocols bgp graceful-restart],
[edit routing-instances *routing-instance-name* protocols bgp graceful-restart]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Specify the maximum time that stale routes are kept during a restart. The **stale-routes-time** statement allows you to set the length of time the routing device waits to receive messages from restarting neighbors before declaring them down.

Options *seconds*—Time the router waits to receive messages from restarting neighbors before declaring them down.

Range: 1 through 600 seconds

Default: 300 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Graceful Restart Options for BGP on page 115
- Configuring Graceful Restart for QFabric Systems
- restart-time (BGP Graceful Restart) on page 169

traceoptions (Protocols)

Syntax

```
traceoptions {
  file name <size size> <files number> <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit protocols isis],
[edit protocols (ospf | ospf3)]

Release Information

Statement introduced before Junos OS Release 7.4.
graceful-restart flag for IS-IS and OSPF/OSPFv3 added in Junos OS Release 8.4.

Description

Define tracing operations that graceful restart functionality in the router.

To specify more than one tracing operation, include multiple **flag** statements.

Default

If you do not include this statement, no global tracing operations are performed.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file ***name***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place global routing protocol tracing output in the file **routing-log**.

files ***number***—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

Range: 2 through 1000 files

Default: 2 files

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

flag ***flag***—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. The nonstop active routing tracing option is:

- **graceful-restart**—Tracing operations for nonstop active routing

no-world-readable—Restrict users from reading the log file.

size ***size***—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 128 KB

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

world-readable—Allow users to read the log file.

Required Privilege Level

routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Tracking Graceful Restart Events on page 120

PART 7

Virtual Router Redundancy Protocol

CHAPTER 17

VRRP Overview

This chapter contains the following section:

Understanding VRRP

For Ethernet, Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, and logical interfaces, you can configure the Virtual Router Redundancy Protocol (VRRP) or VRRP for IPv6. VRRP enables hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts. The VRRP routing platforms share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routing platforms is the master (active) and the others are backups. If the master fails, one of the backup routers becomes the new master router, providing a virtual default routing platform and enabling traffic on the LAN to be routed without relying on a single routing platform. Using VRRP, a backup router can take over a failed default router within a few seconds. This is done with minimum VRRP traffic and without any interaction with the hosts.

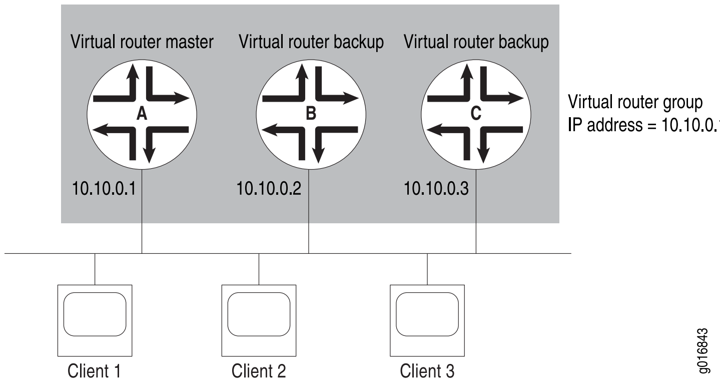
Routers running VRRP dynamically elect master and backup routers. You can also force assignment of master and backup routers using priorities from 1 through 255, with 255 being the highest priority. In VRRP operation, the default master router sends advertisements to backup routers at regular intervals. The default interval is 1 second. If a backup router does not receive an advertisement for a set period, the backup router with the next highest priority takes over as master and begins forwarding packets.

NOTE: **To minimize network traffic, VRRP is designed in such a way that only the router that is acting as the master sends out VRRP advertisements at any given point in time. The backup routers do not send any advertisement until and unless they take over mastership.**

VRRP for IPv6 provides a much faster switchover to an alternate default router than IPv6 Neighbor Discovery (ND) procedures. Typical deployments use only one backup router.

Figure 8 on page 176 illustrates a basic VRRP topology. In this example, Routers A, B, and C are running VRRP and together they make up a virtual router. The IP address of this virtual router is 10.10.0.1 (the same address as the physical interface of Router A).

Figure 8: Basic VRRP



Because the virtual router uses the IP address of the physical interface of Router A, Router A is the master VRRP router, while Routers B and C function as backup VRRP routers. Clients 1 through 3 are configured with the default gateway IP address of 10.10.0.1. As the master router, Router A forwards packets sent to its IP address. If the master virtual router fails, the router configured with the higher priority becomes the master virtual router and provides uninterrupted service for the LAN hosts. When Router A recovers, it becomes the master virtual router again.

VRRP is defined in RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*. VRRP for IPv6 is defined in Internet draft draft-ietf-vrrp-ipv6-spec-08.txt, *Virtual Router Redundancy Protocol for IPv6*. See also Internet draft draft-ietf-vrrp-unified-mib-06.txt, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6*.

NOTE: **Even though VRRP, as defined in RFC 3768, does not support authentication, the Junos OS implementation of VRRP supports authentication as defined in RFC 2338. This support is achieved through the backward compatibility options in RFC 3768.**

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Configuring Basic VRRP Support on page 186

Junos OS Support for VRRPv3

Prior to Junos OS Release 12.2, Junos OS supported RFC 3768, *Virtual Router Redundancy Protocol (VRRP)* and Internet draft draft-ietf-vrrp-ipv6-spec-08, *Virtual Router Redundancy Protocol for IPv6*. Starting with Junos OS Release 12.2, Junos OS supports Virtual Router Redundancy Protocol version 3 (VRRPv3). The support for VRRPv3 is implemented in compliance with RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6*. Junos OS Release 12.2 also supports VRRP MIB for VRRPv3.

The support for VRRP MIB for VRRPv3 is implemented in compliance with RFC 6527, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3)*.

When you configure VRRP for IPv4 or IPv6 networks, you can enable VRRPv3 by configuring the **version-3** statement at the **[edit protocols vrrp]** hierarchy level.

NOTE: When enabling VRRPv3, you must ensure that VRRPv3 is enabled on all the VRRP routers in the network. This is because VRRPv3 does not interoperate with the previous versions of VRRP.

Understanding VRRPv3 Behavioral Differences

You must consider the following aspects when enabling VRRPv3 for your IPv4 or IPv6 networks:

- When VRRP for IPv6 is configured without enabling VRRPv3, the VRRP checksum is calculated according to section 5.3.8 of RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*. However, when VRRPv3 is enabled, the VRRP checksum is calculated according to section 5.3.7 of draft-ietf-vrrp-ipv6-spec-08.txt, *Virtual Router Redundancy Protocol for IPv6*. Therefore, when IPv6 VRRP packets are received or transmitted, the VRRP checksum is calculated according to:
  - RFC 3768, when VRRPv3 is *not* enabled.
  - draft-ietf-vrrp-ipv6-spec-08.txt, when VRRPv3 is enabled.
- The **tcpdump** utility calculates the VRRP checksum according to draft-ietf-vrrp-ipv6-spec-08.txt. Therefore, when **tcpdump** parses IPv6 VRRP packets that are received from older Junos OS releases (prior to Junos OS Release 12.2), the **bad vrrp cksum** message is displayed:

  ```
  23:20:32.657328 Out
  ...
          -----original packet-----
          00:00:5e:00:02:03 > 33:33:00:00:00:12, ethertype IPv6 (0x86dd), length
   94: (class 0xc0, hlim 255, next-header: VRRP (112), length: 40)
  fe80::224:dcff:fe47:57f > ff02::12: VRRPv3-advertisement 40: vrid=3 prio=100
  intvl=100(centisec)  (bad vrrp cksum b4e2!) addrs(2):
  fe80::200:5eff:fe00:3,2001:4818:f000:14::1
                           3333 0000 0012 0000 5e00 0203 86dd 6c00
                           0000 0028 70ff fe80 0000 0000 0000 0224
                           dcff fe47 057f ff02 0000 0000 0000 0000
                           0000 0000 0012 3103 6402 0064 b4e2 fe80
                           0000 0000 0000 0200 5eff fe00 0003 2001
                           4818 f000 0014 0000 0000 0000 0001
  ```

  You can ignore this message because it does not indicate VRRP failure.
- When VRRPv3 is enabled, the **authentication-type** and **authentication-key** statements (for IPv4 VRRP) cannot be configured for any VRRP groups. Therefore, if authentication is required, you need to configure alternative non-VRRP authentication mechanisms.

- When VRRPv3 is enabled, the **advertise-interval** statement (for IPv4 VRRP) and the **inet6-advertise-interval** statement (for IPv6 VRRP) cannot be used to configure advertisement intervals. Instead, use the **fast-interval** statement to configure advertisement intervals.
- VRRPv3 for IPv4 does not interoperate with the previous versions of VRRP. If VRRPv2 IPv4 advertisement packets are received by a router on which VRRPv3 is enabled, the router transitions itself to the backup state to avoid creating multiple masters in the network. Due to this behavior, you must be cautious when enabling VRRPv3 on your existing VRRPv2 networks. See "Understanding VRRPv2 to VRRPv3 Transition" on page 178 for more information.

NOTE: **VRRPv3 advertisement packets are ignored by the routers on which previous versions of VRRP are configured.**

Understanding VRRPv2 to VRRPv3 Transition

You must enable VRRPv3 in your network only if VRRPv3 can be enabled on all the VRRP routers in your network. Even if VRRPv3 can be enabled on all the VRRP routers in your network, care must be taken to avoid traffic loss when you transition your network to VRRPv3. This is because it is practically not possible to configure VRRPv3 on all routers simultaneously. There is a small time frame in the transition period during which VRRPv2 and VRRPv3 coexist in the network. During this period, to avoid having multiple masters in the network, the VRRPv3 IPv4 routers switch to the backup state when they receive a VRRPv2 IPv4 advertisement packet. VRRPv2 IPv4 packets are always given the highest priority. Additionally, to avoid having multiple masters in your IPv6 network due to checksum differences, you need to disable VRRP for IPv6 on the backup routers.

NOTE: **Configuration change from VRRPv2 to VRRPv3 (or VRRPv3 to VRRPv2) restarts the VRRP state machine on all the configured VRRP groups.**

The following example illustrates the steps and events that take place during a VRRPv2 to VRRPv3 transition:

Consider a scenario where two VRRPv2 routers, R1 and R2, are configured in two groups, G1 and G2. The R1 router acts as the master for G1 and the R2 router acts as the master for G2. Table 8 on page 178 lists the transition steps and events for this setup:

Table 8: Example: VRRPv2 to VRRPv3 Transition Steps and Events

1. Upgrade the R1 router with Junos OS Release 12.2 or later.
   - R2 becomes master for both G1 and G2.
   - After the upgrade of the R1 router is completed, R1 becomes the master for G1. R2 remains as the master for G2.
2. Upgrade the R2 router with Junos OS Release 12.2 or later.
   - R1 becomes master for both G1 and G2.
   - After the upgrade of R2 router is completed, R2 becomes the master for G2. R1 remains as the master for G1.

Table 8: Example: VRRPv2 to VRRPv3 Transition Steps and Events *(continued)*

For IPv4	For IPv6
3. Enable VRRPv3 on the R1 router. • Because VRRPv2 IPv4 advertisement packets are given higher priority, R1 becomes the backup for both G1 and G2. 4. Enable VRRPv3 on the R2 router. • R1 becomes the master for G1 and R2 becomes the master for G2.	3. Deactivate the G1 and G2 groups on the R2 router. • G1 and G2 groups on the R1 router become master. 4. Enable VRRPv3 on the R1 router. • R1 becomes master for both G1 and G2. 5. Enable VRRPv3 on the R2 router. 6. Activate G1 and G2 groups on the R2 router. • R2 becomes master for G2. • R1 remains as the master for G1.

Related Documentation

- Understanding High Availability Features on Juniper Networks Routers on page 3
- Configuring Basic VRRP Support on page 186
- VRRP Configuration Hierarchy on page 183
- VRRP for IPv6 Configuration Hierarchy on page 184
- **authentication-type on page 214**
- **authentication-key on page 213**
- **fast-interval on page 217**
- **inet6-advertise-interval on page 220**
- **version-3 on page 231**
- **virtual-link-local-address on page 233**

Improving the Convergence Time for VRRP

You can enable faster convergence time for the configured Virtual Router Redundancy Protocol (VRRP), thereby reducing the traffic restoration time to less than 1 second. To improve the convergence time for the VRRP, perform the following tasks:

- **Configure the distributed periodic packet management process**—When the VRRP process is busy and does not send VRRP advertisements, the backup VRRP routers might assume that the master router is down and take over as the master router, causing unnecessary flaps. To address this problem and to reduce the load on the VRRP process, Junos OS uses the distributed periodic packet management (PPM) process to send VRRP advertisements on behalf of the VRRP process.

  To configure the distributed PPM process, include the **delegate-processing** statement at the **[edit protocols vrrp]** hierarchy level.

- **Disable the skew timer**—The skew timer in VRRP is used to ensure that two backup routers do not switch to the master state at the same time in case of a failover situation. When there is only one master router and one backup router in the network deployment, you can disable the skew timer, thereby reducing the time required to transition to the master state.

To disable the skew timer, include the **skew-timer-disable** statement at the **[edit protocols vrrp]** hierarchy level.

- **Configure the number of fast advertisements that can be missed by a backup router before it starts transitioning to the master state**—The backup router waits until a certain number of advertisement packets are lost after which it transitions to the master state. This waiting time can be fatal in scenarios such as router failure or link failure. To avoid such a situation and to enable faster convergence time, in Junos OS Release 12.2 and later, you can configure a fast advertisement interval value that specifies the number of fast advertisements that can be missed by a backup router before it starts transitioning to the master state.

  To configure the fast advertisement interval, include the **global-advertisements-threshold** statement at the **[edit protocols vrrp]** hierarchy level.

- **Configure inheritance of VRRP groups**—Junos OS enables you to configure VRRP groups on the various subnets of a virtual LAN (VLAN) to inherit the state and configuration of one of the groups, which is known as the active VRRP group. When the **vrrp-inherit-from** statement is included in the configuration, only the active VRRP group, from which the other VRRP groups inherit the state, sends out frequent VRRP advertisements and processes incoming VRRP advertisements. Use inherit groups for scaled configurations. For example, if you have 1000 VRRP groups with an advertisement interval of 100 ms, then use inherit groups.

  To configure inheritance for a VRRP group, include the **vrrp-inherit-from** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]** hierarchy level.

NOTE:

- The reduction in convergence time is not applicable when VRRP is configured over integrated routing and bridging (IRB) interfaces, aggregated Ethernet interfaces, and multichassis link aggregation group (MC-LAG) interfaces.
- Compared to other routers, the convergence time and the traffic restoration time are less for MX Series routers with MPCs.
- Reduction in convergence time is applicable for all types of configurations at the physical interface but the convergence time might not be less than 1 second for all the configurations. The convergence time depends on the number of groups that are transitioning from the backup to the master state and the interval at which these groups are transitioning.

Related Documentation

- Configuring Inheritance for a VRRP Group on page 198
- Configuring VRRP to Improve Convergence Time on page 202
- **delegate-processing on page 216**

- global-advertisements-threshold on page 218
- skew-timer-disable on page 227

CHAPTER 18

VRRP Configuration Guidelines

This chapter contains the following topics:

VRRP Configuration Hierarchy

To configure VRRP for IPv4, include the following statements:

```
vrrp-group group-id {
  (accept-data | no-accept-data);
```

```
    advertise-interval seconds;
    authentication-key key;
    authentication-type authentication;
    fast-interval milliseconds;
    (preempt | no-preempt) {
      hold-time seconds;
    }
    priority number;
    track {
      interface interface-name {
        bandwidth-threshold bits-per-second priority-cost priority;
        priority-cost priority;
      }
      priority-hold-time seconds;
      route prefix/prefix-length routing-instance instance-name priority-cost priority;
    }
    virtual-address [ addresses ];
    vrrp-inherit-from vrrp-group;
  }
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address*]**

If you want to enable Virtual Router Redundancy Protocol version 3 (VRRPv3), configure the **version-3** statement at the **[edit protocols vrrp]** hierarchy level.

NOTE: When enabling VRRPv3, you must ensure that VRRPv3 is enabled on all the VRRP routers in the network. This is because VRRPv3 does not interoperate with the previous versions of VRRP.

Related Documentation

- Understanding VRRP on page 175
- VRRP for IPv6 Configuration Hierarchy on page 184
- Configuring Basic VRRP Support on page 186
- Example: Configuring VRRP on page 203
- Junos OS Support for VRRPv3 on page 176
- version-3 on page 231

VRRP for IPv6 Configuration Hierarchy

To configure VRRP for IPv6, include the following statements:

```
  vrrp-inet6-group group-id {
    (accept-data | no-accept-data);
    fast-interval milliseconds;
    inet6-advertise-interval seconds;
```



```
    (preempt | no-preempt) {
      hold-time seconds;
    }
    priority number;
    track {
      interface interface-name {
        bandwidth-threshold bits-per-second priority-cost priority;
        priority-cost priority;
      }
      priority-hold-time seconds;
      route prefix/prefix-length routing-instance instance-name priority-cost priority;
    }
    virtual-inet6-address [ addresses ];
    virtual-link-local-address ipv6-address;
  }
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address*]**

If you want to enable Virtual Router Redundancy Protocol version 3 (VRRPv3), configure the **version-3** statement at the **[edit protocols vrrp]** hierarchy level.

NOTE: When enabling VRRPv3, you must ensure that VRRPv3 is enabled on all the VRRP routers in the network. This is because VRRPv3 does not interoperate with the previous versions of VRRP.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- Example: Configuring VRRP for IPv6 on page 205
- Junos OS Support for VRRPv3 on page 176
- **version-3 on page 231**

Configuring the Startup Period for VRRP Operations

To configure the startup period for VRRP operations, include the **startup-silent-period** statement at the **[edit protocols vrrp]** hierarchy level:

```
[edit protocols vrrp]
startup-silent-period seconds;
```

NOTE: During the silent startup period, the **show vrrp detail** command output shows a value of 0 for **Master priority**, and your own IP address for **Master router**. These values indicate that the Master selection is not completed yet, and these values can be ignored.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- Configuring Basic VRRP Support on page 186
- Configuring VRRP Authentication (IPv4 Only) on page 188
- Example: Configuring VRRP on page 203

Configuring Basic VRRP Support

An interface can be a member of one or more VRRP groups. To configure basic VRRP support, configure VRRP groups on interfaces by including the **vrrp-group** statement:

```
vrrp-group group-id {
  priority number;
  virtual-address [ addresses ];
}
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address*]**

To configure basic VRRP for IPv6 support, configure VRRP group support on interfaces by including the **vrrp-inet6-group** statement:

```
vrrp-inet6-group group-id {
  priority number;
  virtual-inet6-address [ addresses ];
  virtual-link-local-address ipv6-address;
}
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address*]**

Within a VRRP group, the master virtual router and the backup virtual router must be configured on two different routing platforms.

For each VRRP group, you must configure the following:

- Group identifier—Assign a value from 0 through 255.
- Address of one or more virtual routers that are members of the VRRP group—Normally, you configure only one virtual IP address per group. However, you can configure up to eight addresses. Do not include a prefix length in a virtual IP address. The following considerations apply to configuring a virtual IP address:

  - The virtual router IP address must be the same for all routing platforms in the VRRP group.
  - If you configure a virtual IP address to be the same as the physical interface's address, the interface becomes the master virtual router for the group. In this case, you must configure the priority to be 255, and you must configure preemption by including the **preempt** statement.
  - If the virtual IP address you choose is not the same as the physical interface's address, you must ensure that the virtual IP address does not appear anywhere else in the routing platform's configuration. Verify that you do not use this address for other interfaces, for the IP address of a tunnel, or for the IP address of static ARP entries.
  - You cannot configure a virtual IP address to be the same as the interface's address for an aggregated Ethernet interface. This configuration is not supported.
  - For VRRP for IPv6, the EUI-64 option cannot be used. In addition, the Duplicate Address Detection (DAD) process will not run for virtual IPv6 addresses.
  - You cannot configure the same virtual IP address on interfaces that belong to the same logical system and routing instance combination. However, you can configure the same virtual IP address on interfaces that belong to different logical system and routing instance combinations.

- Virtual link-local address—(VRRP for IPv6 only) You must explicitly define a virtual link-local address for each VRRP for IPv6 group. Otherwise, when you attempt to commit the configuration, the commit request fails. The virtual link-local address must be on the same subnet as the physical interface address.

- Priority for this routing platform to become the master virtual router—Configure the value used to elect the master virtual router in the VRRP group. It can be a number from 1 through 255. The default value for backup routers is 100. A larger value indicates a higher priority. The routing platform with the highest priority within the group becomes the master router.

  NOTE: If there are two or more backup routers with the same priority, the router that has the highest primary address becomes the master.

NOTE: Mixed tagging (configuring two logical interfaces on the same Ethernet port, one with single-tag framing and one with dual-tag framing) is supported only for interfaces on Gigabit Ethernet IQ2 and IQ PICs. If you include the **flexible-vlan-tagging** statement at the **[edit interfaces *interface-name*]** hierarchy level for a VRRP-enabled interface on a PIC that does not support mixed tagging, VRRP on that interface is disabled. In the output of the **show vrrp summary** command, the interface status is listed as **Down**.

NOTE: If you enable MAC source address filtering on an interface, you must include the virtual MAC address in the list of source MAC addresses that you specify in the **source-address-filter** statement at the **[edit interfaces *interface-name*]** hierarchy level. (For more information, see the Junos® OS Network Interfaces.) MAC addresses ranging from **00:00:5e:00:01:00** through **00:00:5e:00:01:ff** are reserved for VRRP, as defined in RFC 2378. The VRRP group number must be the decimal equivalent of the last hexadecimal byte of the virtual MAC address.

Related Documentation

- Understanding VRRP on page 175
- Junos OS Support for VRRPv3 on page 176
- VRRP Configuration Hierarchy on page 183
- Configuring the Startup Period for VRRP Operations on page 185
- Configuring VRRP Authentication (IPv4 Only) on page 188
- Configuring the Advertisement Interval for the VRRP Master Router on page 189
- Example: Configuring VRRP on page 203

Configuring VRRP Authentication (IPv4 Only)

VRRP (IPv4 only) protocol exchanges can be authenticated to guarantee that only trusted routing platforms participate in routing in an autonomous system (AS). By default, VRRP authentication is disabled. You can configure one of the following authentication methods. Each VRRP group must use the same method.

- Simple authentication—Uses a text password included in the transmitted packet. The receiving routing platform uses an authentication key (password) to verify the packet.
- Message Digest 5 (MD5) algorithm—Creates the authentication data field in the IP authentication header. This header is used to encapsulate the VRRP PDU. The receiving routing platform uses an authentication key (password) to verify the authenticity of the IP authentication header and VRRP PDU.

To enable authentication and specify an authentication method, include the **authentication-type** statement:

```
authentication-type authentication;
```

authentication can be **simple** or **md5**. The authentication type must be the same for all routing platforms in the VRRP group.

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**

- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**

If you include the **authentication-type** statement, you can configure a key (password) on each interface by including the **authentication-key** statement:

authentication-key *key*;

key (the password) is an ASCII string. For simple authentication, it can be from 1 through 8 characters long. For MD5 authentication, it can be from 1 through 16 characters long. If you include spaces, enclose all characters in quotation marks (" "). The key must be the same for all routing platforms in the VRRP group.

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**

NOTE: When VRRPv3 is enabled, the **authentication-type** and **authentication-key** statements cannot be configured for any VRRP groups. Therefore, if authentication is required, you need to configure alternative non-VRRP authentication mechanisms.

Related Documentation

- Understanding VRRP on page 175
- Junos OS Support for VRRPv3 on page 176
- VRRP Configuration Hierarchy on page 183
- Configuring Basic VRRP Support on page 186
- Example: Configuring VRRP on page 203

Configuring the Advertisement Interval for the VRRP Master Router

By default, the master router sends VRRP advertisement packets every second to all members of the VRRP group. These packets indicate that the master router is still operational. If the master router fails or becomes unreachable, the backup router with the highest priority value becomes the new master router.

You can modify the advertisement interval in seconds or in milliseconds. The interval must be the same for all routing platforms in the VRRP group.

For VRRP for IPv6, you must configure IPv6 router advertisements for the interface on which VRRP is configured to send IPv6 router advertisements for the VRRP group. To do so, include the **interface *interface-name*** statement at the **[edit protocols router-advertisement]** hierarchy level. (For information about this statement and guidelines, see the Junos OS Routing Protocols Configuration Guide.) When an interface receives an IPv6 router solicitation message, it sends an IPv6 router advertisement to all VRRP groups configured on it. In the case of logical systems, IPv6 router advertisements are not sent to VRRP groups.

NOTE: The master VRRP for an IPv6 router must respond to a router solicitation message with the virtual IP address of the router. However, when the **interface** ***interface-name*** statement is included at the **[edit protocols router-advertisement]** hierarchy level, the backup VRRP for an IPv6 router might send a response before the VRRP master responds, so that the default route of the client is not set to the master VRRP router's virtual IP address. To avoid this situation, include the **virtual-router-only** statement at the **[edit protocols router-advertisement interface *interface-name*]** hierarchy level. When this statement is included, router advertisements are sent only for VRRP IPv6 groups configured on the interface (if the groups are in the master state). You must include this statement on both the master and backup VRRP for IPv6 routers.

This topic contains the following sections:

Modifying the Advertisement Interval in Seconds

To modify the time, in seconds, between the sending of VRRP advertisement packets, include the **advertise-interval** statement:

```
advertise-interval seconds;
```

The interval can be from 1 through 255 seconds.

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**

NOTE: When VRRPv3 is enabled, the **advertise-interval** statement cannot be used to configure advertisement intervals. Instead, use the **fast-interval** statement to configure advertisement intervals.

Modifying the Advertisement Interval in Milliseconds

To modify the time, in milliseconds, between the sending of VRRP advertisement packets, include the **fast-interval** statement:

```
fast-interval milliseconds;
```

The interval can be from 10 through 40,950 milliseconds.

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]**

NOTE: In the VRRP PDU, Junos OS sets the advertisement interval to 0. When you configure VRRP with other vendors' routers, the **fast-interval** statement works correctly only when the other routers also have an advertisement interval set to 0 in the VRRP PDUs. Otherwise, Junos OS interprets other routers' settings as advertisement timer errors.

To modify the time, in milliseconds, between the sending of VRRP for IPv6 advertisement packets, include the **inet6-advertise-interval** statement:

```
inet6-advertise-interval ms;
```

The range of values is from 100 through 40,950 milliseconds (ms).

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]**

NOTE: When VRRPv3 is enabled, the **inet6-advertise-interval** statement cannot be used to configure advertisement intervals. Instead, use the **fast-interval** statement to configure advertisement intervals.

Related Documentation

- Understanding VRRP on page 175
- Junos OS Support for VRRPv3 on page 176
- VRRP Configuration Hierarchy on page 183
- Configuring Basic VRRP Support on page 186
- Configuring a Backup Router to Preempt the Master Router on page 192
- Modifying the Preemption Hold-Time Value on page 192
- Configuring Asymmetric Hold Time for VRRP Routers on page 193
- Configuring the Silent Period on page 200
- Example: Configuring VRRP on page 203

Configuring a Backup Router to Preempt the Master Router

By default, a higher-priority backup router preempts a lower-priority master router. To explicitly enable the master router to be preempted, include the **preempt** statement:

```
preempt;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (Inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]**

To prohibit a higher-priority backup router from preempting a lower-priority master router, include the **no-preempt** statement:

```
no-preempt;
```

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- Configuring the Advertisement Interval for the VRRP Master Router on page 189
- Modifying the Preemption Hold-Time Value on page 192
- Configuring Asymmetric Hold Time for VRRP Routers on page 193
- Example: Configuring VRRP on page 203

Modifying the Preemption Hold-Time Value

The hold time is the maximum number of seconds that can elapse before a higher-priority backup router preempts the master router. You might want to configure a hold time so that all Junos OS components converge before preemption.

By default, the hold-time value is 0 seconds. A value of 0 means that preemption can occur immediately after the backup router comes online. Note that the hold time is counted from the time the backup router comes online. The hold time is only valid when the VRRP router is just coming online.

To modify the preemption hold-time value, include the **hold-time** statement:

```
hold-time seconds;
```

The hold time can be from 0 through 3600 seconds.

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id* preempt]**

- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (Inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id* preempt]**

Related Documentation

- VRRP Configuration Hierarchy on page 183
- Configuring the Advertisement Interval for the VRRP Master Router on page 189
- Configuring a Backup Router to Preempt the Master Router on page 192
- Configuring Asymmetric Hold Time for VRRP Routers on page 193
- Example: Configuring VRRP on page 203

Configuring Asymmetric Hold Time for VRRP Routers

In Junos OS Release 9.5 and later, the **asymmetric-hold-time** statement at the **[edit protocols vrrp]** hierarchy level enables you to configure a VRRP master router to switch over to the backup router immediately—that is, without waiting for the priority hold time to expire—when a tracked interface or route goes down or when the bandwidth of a tracked interface decreases. Such events can cause an immediate reduction in the priority based on the configured priority cost for the event, and trigger a mastership election.

However, when the tracked route or interface comes up again, or when the bandwidth for a tracked interface increases, the backup (original master) router waits for the hold time to expire before it updates the priority and initiates the switchover if the priority is higher than the priority for the VRRP master (original backup) router.

If the **asymmetric-hold-time** statement is not configured, the VRRP master waits for the hold time to expire before it initiates a switchover when a tracked route goes down.

Example: Configuring Asymmetric Hold Time

```
[edit]
user@host# set protocols vrrp asymmetric-hold-time
[edit]
user@host# show protocols vrrp
asymmetric-hold-time;
```

Related Documentation

- VRRP Configuration Hierarchy on page 183
- Configuring the Advertisement Interval for the VRRP Master Router on page 189
- Configuring a Backup Router to Preempt the Master Router on page 192
- Modifying the Preemption Hold-Time Value on page 192
- Example: Configuring VRRP on page 203

Configuring an Interface to Accept Packets Destined for the Virtual IP Address

In VRRP implementations where the router acting as the master router is not the IP address owner—the IP address owner is the router that has the interface whose actual IP address is used as the virtual router's IP address (virtual IP address)— the master router accepts only the ARP packets from the packets that are sent to the virtual IP address. Junos OS enables you to override this limitation with the help of the **accept-data** configuration. When the **accept-data** statement is included in the configuration, the master router accepts all packets sent to the virtual IP address even when the master router is not the IP address owner.

NOTE: If the master router is the IP address owner or has its priority set to 255, the master router, by default, accepts all packets addressed to the virtual IP address. In such cases, the **accept-data** configuration is not required.

To configure an interface to accept all packets sent to the virtual IP address, include the **accept-data** statement:

```
accept-data;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (Inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]

To prevent a master router that is the IP address owner or has its priority set to 255 from accepting packets other than the ARP packets addressed to the virtual IP address, include the **no-accept-data** statement:

```
no-accept-data;
```

NOTE:

- If you want to restrict the incoming IP packets to ICMP packets only, you must configure firewall filters to accept only ICMP packets.
- If you include the **accept-data** statement, your routing platform configuration does not comply with RFC 3768 (see section 6.4.3 of RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- Example: Configuring VRRP on page 203

Configuring a Logical Interface to Be Tracked

VRRP can track whether a logical interface is up, down, or not present, and can also dynamically change the priority of the VRRP group based on the state of the tracked logical interface, triggering a new master router election. VRRP can also track the operational speed of a logical interface and dynamically update the priority of the VRRP group when the speed crosses a configured threshold.

When interface tracking is enabled, you cannot configure a priority of 255 (a priority of 255 designates the master router). For each VRRP group, you can track up to 10 logical interfaces.

To configure a logical interface to be tracked, include the following statements:

```
track {
  interface interface-name {
    bandwidth-threshold bits-per-second priority-cost priority;
    priority-cost priority;
  }
  priority-hold-time seconds;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**
- **[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]**

The interface specified is the interface to be tracked for the VRRP group. The priority hold time is the minimum length of time that must elapse between dynamic priority changes. A tracking event, such as an interface state change (up or down) or a change in bandwidth, triggers one of the following responses:

- The first tracking event initiates the priority hold timer, and also initializes the pending priority based on the current priority and the priority cost. However, the current priority remains unchanged.
- A tracking event or a manual configuration change that occurs while the priority hold timer is on triggers a pending priority update. However, the current priority remains unchanged.

This ensures that Junos OS does not initiate mastership elections every time a tracked interface flaps.

When the priority hold time expires, the current priority inherits the value from the pending priority, and the pending priority ceases.

NOTE: If you have configured **asymmetric-hold-time**, VRRP does not wait for the priority hold time to expire before initiating mastership elections if a tracked interface fails (state changes from **up** to **down**), or if the available bandwidth for a tracked interface decreases. For more information about **asymmetric-hold-time**, see "Configuring Asymmetric Hold Time for VRRP Routers" on page 193.

The bandwidth threshold specifies a threshold for the tracked interface. When the bandwidth of the tracked interface drops below the configured bandwidth threshold value, the VRRP group uses the bandwidth threshold priority cost. You can track up to five bandwidth threshold statements for each tracked interface.

The priority cost is the value to be subtracted from the configured VRRP priority when the tracked logical interface goes down, forcing a new master router election. The value can be from 1 through 254. The sum of the costs for all tracked logical interfaces or routes must be less than or equal to the configured priority of the VRRP group.

If you are tracking more than one interface, the router applies the sum of the priority costs for the tracked interfaces (at most, only one priority cost for each tracked interface) to the VRRP group priority. However, the interface priority cost and bandwidth threshold priority cost values for each VRRP group are not cumulative. The router uses only one priority cost to a tracked interface as indicated in Table 9 on page 196:

Table 9: Interface State and Priority Cost Usage

Tracked Interface State	Priority Cost Usage
Down	**priority-cost *priority***
Not down; media speed below one or more bandwidth thresholds	Priority-cost of the lowest applicable bandwidth threshold

You must configure an interface priority cost only if you have configured no bandwidth thresholds. If you have not configured an interface priority cost value, and the interface is down, the interface uses the bandwidth threshold priority cost value of the lowest bandwidth threshold.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- Configuring a Route to Be Tracked on page 197
- Example: Configuring VRRP on page 203

Configuring a Route to Be Tracked

VRRP can track whether a route is reachable (that is, the route exists in the routing table of the routing instance included in the configuration) and dynamically change the priority of the VRRP group based on the reachability of the tracked route, triggering a new master router election.

To configure a route to be tracked, include the following statements:

```
track {
  priority-hold-time seconds;
  route prefix/prefix-length routing-instance instance-name priority-cost priority;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**
- **[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]**

The route prefix specified is the route to be tracked for the VRRP group. The priority hold time is the minimum length of time that must elapse between dynamic priority changes. A route tracking event, such as adding a route to or removing a route from the routing table, might trigger one or more of the following:

- The first tracking event initiates the priority hold timer, and also initializes the pending priority based on the current priority and the priority cost. However, the current priority remains unchanged.
- A tracking event or a manual configuration change that occurs while the priority hold timer is on triggers a pending priority update. However, the current priority remains unchanged.

When the priority hold time expires, the current priority inherits the value from the pending priority, and the pending priority ceases.

This ensures that Junos OS does not initiate mastership elections every time a tracked route flaps.

NOTE: If you have configured **asymmetric-hold-time**, VRRP does not wait for the priority hold time to expire before initiating mastership elections if a tracked route is removed from the routing table. For more information about **asymmetric-hold-time**, see "Configuring Asymmetric Hold Time for VRRP Routers" on page 193.

The routing instance is the routing instance in which the route is to be tracked. If the route is in the default, or global, routing instance, specify the instance name as **default**.

NOTE: **Tracking a route that belongs to a routing instance from a different logical system is not supported.**

The priority cost is the value to be subtracted from the configured VRRP priority when the tracked route goes down, forcing a new master router election. The value can be from 1 through 254. The sum of the costs for all tracked logical interfaces or routes must be less than or equal to the configured priority of the VRRP group.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- Configuring a Logical Interface to Be Tracked on page 195
- Example: Configuring VRRP Route Tracking on page 206

Configuring Inheritance for a VRRP Group

Junos OS enables you to configure VRRP groups on the various subnets of a VLAN to inherit the state and configuration of one of the groups, which is known as the *active VRRP group*. When the **vrrp-inherit-from** configuration statement is included in the configuration, only the active VRRP group, from which the other VRRP groups are inheriting the state, sends out frequent VRRP advertisements, and processes incoming VRRP advertisements. The groups that are inheriting the state do not process any incoming VRRP advertisement because the state is always inherited from the active VRRP group. However, the groups that are inheriting the state do send out VRRP advertisements once every 2 to 3 minutes to facilitate MAC address learning on the switches placed between the VRRP routers.

If the **vrrp-inherit-from** statement is not configured, each of the VRRP master groups in the various subnets on the VLAN sends out separate VRRP advertisements and adds to the traffic on the VLAN.

To configure inheritance for a VRRP group, include the **vrrp-inherit-from** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]** hierarchy level.

```
[edit interfaces interface-name unit logical-unit-number family inet address address
  vrrp-group group-id]
vrrp-inherit-from vrrp-group;
```

When you configure a group to inherit a state from another group, the inheriting groups and the active group must be on the same physical interface and logical system. However, the groups do not need to necessarily be on the same routing instance (as was in Junos OS releases earlier than 9.6), VLAN, or logical interface.

When you include the **vrrp-inherit-from** statement for a VRRP group, the VRRP group inherits the following parameters from the active group:

- **advertise-interval**
- **authentication-key**
- **authentication-type**
- **fast-interval**
- **preempt | no-preempt**
- **priority**
- **track interfaces**
- **track route**

However, you can configure the **accept-data | no-accept-data** statement for the group to specify whether the interface should accept packets destined for the virtual IP address.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183

Tracing VRRP Operations

To trace VRRP operations, include the **traceoptions** statement at the **[edit protocols vrrp]** hierarchy level.

By default, VRRP logs the error, data carrier detect (DCD) configuration, and routing socket events in a file in the **/var/log** directory. By default, this file is named **/var/log/vrrpd**. The default file size is 1 megabyte (MB), and three files are created before the first one gets overwritten.

To change the configuration of the logging file, include the **traceoptions** statement at the **[edit protocols vrrp]** hierarchy level:

```
[edit protocols vrrp]
traceoptions {
  file filename <files number> <match regular-expression> <microsecond-stamp>
    <size size> <world-readable | no-world-readable>;
  flag flag;
  no-remote-trace;
}
flag flag;
```

You can specify the following VRRP tracing flags:

- **all**—Trace all VRRP operations.
- **database**—Trace all database changes.
- **general**—Trace all general events.
- **interfaces**—Trace all interface changes.

- **normal**—Trace all normal events.
- **packets**—Trace all packets sent and received.
- **state**—Trace all state transitions.
- **timer**—Trace all timer events.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183

Configuring the Silent Period

The silent period starts when the interface state is changed from down to up. During this period, the Master Down Event is ignored. Configure the silent period interval to avoid alarms caused by the delay or interruption of the incoming VRRP advertisement packets during the interface startup phase.

To configure the silent period interval that the Master Down Event timer ignores, include the **startup-silent-period** statement at the **[edit protocols vrrp]** hierarchy level:

```
[edit protocols vrrp]
startup-silent-period seconds;
```

NOTE: **During the silent startup period, the show vrrp detail command output shows a value of 0 for Master priority and your IP address for Master router. These values indicate that the Master selection is not completed yet, and these values can be ignored.**

When you have configured **startup-silent-period**, the Master Down Event is ignored until the **startup-silent-period** expires.

For example, configure a VRRP group, *vrrp-group1*, with an advertise interval of 1 second, startup silent period of 10 seconds, and an interface *interface1* with a priority less than 255.

When *interface1* transitions from down to up:

- The *vrrp-group1* group moves to the backup state, and starts the Master Down Event timer (3 seconds; three times the value of the advertise interval, which is 1 second in this case).
- If no VRRP PDU is received during the 3-second period, the **startup-silent-period** (10 seconds in this case) is checked, and if the startup silent period has not expired, the Master Down Event timer is restarted. This is repeated until the **startup-silent-period** expires. In this example, the Master Down Event timer runs four times (12 seconds) by the time the 10-second startup silent period expires.
- If no VRRP PDU is received by the end of the fourth 3-second cycle, *vrrp-group1* takes over mastership.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- **startup-silent-period on page 227**

Configuring Passive ARP Learning for Backup VRRP Routers

By default, the backup VRRP router drops ARP requests for the VRRP-IP to VRRP-MAC address translation. This means that the backup router does not learn the ARP (IP-to-MAC address) mappings for the hosts sending the requests. When it detects a failure of the master router and transitions to become the new master router, the backup router must learn all the entries that were present in the ARP cache of the master router. In environments with many directly attached hosts, such as metro Ethernet environments, the number of ARP entries to learn can be high. This can cause a significant transition delay, during which the traffic transmitted to some of the hosts might be dropped.

Passive ARP learning enables the ARP cache in the backup router to hold approximately the same contents as the ARP cache in the master router, thus preventing the problem of learning ARP entries in a burst. To enable passive ARP learning, include the **passive-learning** statement at the **[edit system arp]** hierarchy level:

```
[edit system arp]
passive-learning;
```

We recommend setting passive learning on both the backup and master VRRP routers. Doing so prevents the need to manually intervene when the master router becomes the backup router. While a router is operating as the master router, the passive learning configuration has no operational impact. The configuration takes effect only when the router is operating as a backup router.

For information about configuring gratuitous ARP and the ARP aging timer, see the Junos OS System Basics Configuration Guide.

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183

Enabling the Distributed Periodic Packet Management Process for VRRP

Typically, VRRP advertisements are sent by the VRRP process (vrrpd) on the master VRRP router at regular intervals to let other members of the group know that the VRRP master router is operational.

When the vrrpd process is busy and does not send VRRP advertisements, the backup VRRP routers might assume that the master router is down and take over as the master router, causing unnecessary flaps. This takeover might occur even though the original master router is still active and available, and might resume sending advertisements after the traffic has decreased. To address this problem and to reduce the load on the vrrpd process, Junos OS uses the periodic packet management process (ppmd) to send VRRP advertisements on behalf of the vrrpd process. However, you can further delegate the job of sending VRRP advertisements to the distributed ppmd process that resides on the Packet Forwarding Engine.

The ability to delegate the sending of VRRP advertisements to the distributed ppmd process ensures that the VRRP advertisements are sent even when the ppmd process—which is now responsible for sending VRRP advertisements—is busy. Such delegation prevents the possibility of false alarms when the ppmd process is busy. The ability to delegate the sending of VRRP advertisements to distributed ppmd also adds to scalability because the load is shared across multiple ppmd instances and is not concentrated on any single unit

NOTE: CPU-intensive VRRP advertisements, such as advertisements with MD5 authentication or those sent and received over logical interfaces, such as Aggregated Ethernet interfaces, continue to be processed by the VRRP process on the Routing Engine even when distributed ppmd is enabled.

To configure the distributed ppmd process to send VRRP advertisements, include the **delegate-processing** statement at the **[edit protocols vrrp]** hierarchy level:

```
[edit protocols vrrp]
delegate-processing;
```

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183

Configuring VRRP to Improve Convergence Time

You can enable faster convergence time for the configured Virtual Router Redundancy Protocol (VRRP), thereby reducing the traffic restoration time to less than 1 second. To improve the convergence time for VRRP, perform the following tasks.

Before you begin, configure VRRP. See “Example: Configuring VRRP” on page 203.

1. Configure the distributed periodic packet management (PPM) process to send VRRP advertisements when the VRRP process is busy.

   ```
   [edit]
   user@host# set protocols vrrp delegate-processing
   ```

2. Disable the skew timer to reduce the time required to transition to the master state.

   ```
   [edit]
   user@host# set protocols vrrp skew-timer-disable
   ```

   NOTE: When there is only one master router and one backup router in the network deployment, you can disable the skew timer, thereby reducing the time required to transition to the master state.

3. Configure the number of fast advertisements that can be missed by a backup router before it starts transitioning to the master state.

   [edit]
   user@host# **set protocols vrrp global-advertisement-threshold *advertisement-value***

4. Configure VRRP groups on the various subnets of a VLAN to inherit the state and to configure one of the groups.

   [edit]
   user@host# **set interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id***

5. Verify the configuration in operational mode.

   [edit]
   user@host> **show protocols vrrp**

NOTE:

- The reduction in convergence time is not applicable when VRRP is configured over integrated routing and bridging (IRB) interfaces, aggregated Ethernet interfaces, and multichassis link aggregation group (MC-LAG) interfaces.
- Compared to other routers, the convergence time and the traffic restoration time are less for MX Series routers with MPCs.
- Reduction in convergence time is applicable for all types of configurations at the physical interface, but the convergence time might not be less than 1 second for all the configurations. The convergence time depends on the number of groups that are transitioning from the backup to the master state and the interval at which these groups are transitioning.

Related Documentation

- Improving the Convergence Time for VRRP on page 179
- Configuring Inheritance for a VRRP Group on page 198
- **delegate-processing on page 216**
- **global-advertisements-threshold on page 218**
- **skew-timer-disable on page 227**

Example: Configuring VRRP

Configure one master (Router A) and one backup (Router B) routing platform. The address configured in the **virtual-address** statements differs from the addresses configured in the **address** statements. When you configure multiple VRRP groups on an interface, you configure one to be the master virtual router for that group.

On Router A

```
[edit interfaces]
ge-0/0/0 {
  unit 0 {
    family inet {
```

```
            address 192.168.1.20/24 {
               vrrp-group 27 {
                  virtual-address 192.168.1.15;
                  priority 254;
                  authentication-type simple;
                  authentication-key booJUM;
               }
            }
         }
      }
   }
```

On Router B

```
[edit interfaces]
ge-4/2/0 {
   unit 0 {
      family inet {
         address 192.168.1.24/24 {
            vrrp-group 27 {
               virtual-address 192.168.1.15;
               priority 200;
               authentication-type simple;
               authentication-key booJUM;
            }
         }
      }
   }
}
```

Configuring One Router to Be the Master Virtual Router for the Group

```
[edit interfaces]
ge-0/0/0 {
   unit 0 {
      family inet {
         address 192.168.1.20/24 {
            vrrp-group 2 {
               virtual-address 192.168.1.20;
               priority 255;
               advertise-interval 3;
               preempt;
            }
            vrrp-group 10 {
               virtual-address 192.168.1.55;
               priority 201;
               advertise-interval 3;
            }
            vrrp-group 1 {
               virtual-address 192.168.1.54;
               priority 22;
               advertise-interval 4;
            }
         }
      }
   }
}
```


Configuring VRRP and MAC Source Address Filtering

The VRRP group number is the decimal equivalent of the last byte of the virtual MAC address.

```
[edit interfaces]
ge-5/2/0 {
  gigether-options {
    source-filtering;
    source-address-filter {
      00:00:5e:00:01:0a; # Virtual MAC address
    }
  }
  unit 0 {
    family inet {
      address 192.168.1.10/24 {
        vrrp-group 10;  # VRRP group number
        virtual-address 192.168.1.10;
        priority 255;
        preempt;
      }
    }
  }
}
```

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- VRRP for IPv6 Configuration Hierarchy on page 184
- Example: Configuring VRRP for IPv6 on page 205
- Example: Configuring VRRP Route Tracking on page 206

Example: Configuring VRRP for IPv6

Configure VRRP properties for IPv6 in one master (Router A) and one backup (Router B).

On Router A

```
[edit interfaces]
ge-1/0/0 {
  unit 0 {
    family inet6 {
      address fe80::5:0:0:6/64;
      address fec0::5:0:0:6/64 {
        vrrp-inet6-group 3; # VRRP inet6 group number
        virtual-inet6-address fec0::5:0:0:7;
        virtual-link-local-address fe80::5:0:0:7;
        priority 200;
        preempt;
      }
    }
  }
}

[edit protocols]
router-advertisement {
```



```
    interface ge-1/0/0.0 {
      prefix fec0::/64;
      max-advertisement-interval 4;
      }
    }
```

On Router B

```
[edit interfaces]
ge-1/0/0 {
  unit 0 {
    family inet6 {
      address fe80::5:0:0:8/64;
      address fec0::5:0:0:8/64 {
        vrrp-inet6-group 3; # VRRP inet6 group number
        virtual-inet6-address fec0::5:0:0:7;
        virtual-link-local-address fe80::5:0:0:7;
        priority 100;
        preempt;
      }
    }
  }
}

[edit protocols]
router-advertisement {
  interface ge-1/0/0.0 {
    prefix fec0::/64;
    max-advertisement-interval 4;
    }
  }
```

Related Documentation

- Understanding VRRP on page 175
- VRRP Configuration Hierarchy on page 183
- VRRP for IPv6 Configuration Hierarchy on page 184
- Example: Configuring VRRP on page 203
- Example: Configuring VRRP Route Tracking on page 206

Example: Configuring VRRP Route Tracking

Configure Routers R1 and R2 to run VRRP. Configure static routes and a policy for exporting the static routes on Router R3. The VRRP routing instances on R2 track the routes that are advertised by R3.

On Router R1

```
[edit interfaces]
ge-1/0/3 {
  unit 0 {
    vlan-id 1;
    family inet {
      address 200.100.50.2/24 {
        vrrp-group 0 {
          virtual-address 200.100.50.101;
          priority 195;
        }
```

```
                }
            }
        }
    }
```

On Router R2

```
[edit interfaces]
ge-1/0/1 {
    unit 0 {
        vlan-id 1;
        family inet {
            address 200.100.50.1/24 {
                vrrp-group 0 {
                    virtual-address 200.100.50.101;
                    priority 200;
                    track {
                        route 59.0.58.153/32 routing-instance default priority-cost 5;
                        route 59.0.58.154/32 routing-instance default priority-cost 5;
                        route 59.0.58.155/32 routing-instance default priority-cost 5;
                    }
                }
            }
        }
    }
}
```

On Router R3

```
[edit]
policy-options {
    policy-statement static-policy {
        term term1 {
            then accept;
        }
    }
}
protocols {
    ospf {
        export static-policy;
        reference-bandwidth 4g;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
}
routing-options {
    static {
        route 59.0.0.153/32 next-hop 45.45.45.46;
        route 59.0.0.154/32 next-hop 45.45.45.46;
        route 59.0.0.155/32 next-hop 45.45.45.46;
    }
}
```

Related Documentation

- Understanding VRRP on page 175

- VRRP Configuration Hierarchy on page 183
- VRRP for IPv6 Configuration Hierarchy on page 184
- Configuring a Route to Be Tracked on page 197
- Example: Configuring VRRP on page 203
- Example: Configuring VRRP for IPv6 on page 205

CHAPTER 19

Summary of VRRP Configuration Statements

This chapter provides a reference for each of the VRRP configuration statements. The statements are organized alphabetically.

accept-data

Syntax

(accept-data | no-accept-data);

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description

In a Virtual Router Redundancy Protocol (VRRP) configuration, determine whether or not a router that is acting as the master router accepts all packets destined for the virtual IP address.

- **accept-data**—Enable the master router to accept all packets destined for the virtual IP address.
- **no-accept-data**—Prevent the master router from accepting packets other than the ARP packets destined for the virtual IP address.

Default

If the router acting as the master router is the IP address owner or has its priority set to 255, the master router, by default, responds to all packets sent to the virtual IP address. However, if the router acting as the master router does not own the IP address or has its priority set to a value less than 255, the master router responds only to ARP requests.

NOTE:

- If you want to restrict the incoming IP packets to ICMP packets only, you must configure firewall filters to accept only ICMP packets.
- If you include the **accept-data** statement, your routing platform configuration does not comply with RFC 3768 (see section 6.4.3 of RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*.

Required Privilege Level

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

Related Documentation

- Configuring an Interface to Accept Packets Destined for the Virtual IP Address on page 194

advertise-interval

Syntax advertise-interval *seconds*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Configure the interval between Virtual Router Redundancy Protocol (VRRP) IPv4 advertisement packets.

All routers in the VRRP group must use the same advertisement interval.

NOTE: When VRRPv3 is enabled, the **advertise-interval** statement cannot be used to configure advertisement intervals. Instead, use the **fast-interval** statement to configure advertisement intervals.

Options *seconds*—Interval between advertisement packets.

Range: 1 through 255 seconds

Default: 1 second

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring the Advertisement Interval for the VRRP Master Router on page 189
- fast-interval on page 217
- inet6-advertise-interval on page 220
- version-3 on page 231

asymmetric-hold-time

Syntax asymmetric-hold-time;

Hierarchy Level [edit protocols vrrp]

Release Information Statement introduced in Junos OS Release 9.5.

Description Enable the VRRP master router to switch over to the backup router immediately, without waiting for the priority hold time to expire, when a route goes down. However, when the route comes back online, the backup router that is acting as the master waits for the priority hold time to expire before switching the mastership back to the original master VRRP router.

Default **asymmetric-hold-time** is disabled.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Asymmetric Hold Time for VRRP Routers on page 193

authentication-key

Syntax authentication-key *key*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Configure a Virtual Router Redundancy Protocol (VRRP) IPv4 authentication key. You also must specify a VRRP authentication scheme by including the **authentication-type** statement.

All routers in the VRRP group must use the same authentication scheme and password.

NOTE: When VRRPv3 is enabled, the **authentication-type** and **authentication-key** statements cannot be configured for any VRRP groups.

Options *key*—Authentication password. For simple authentication, it can be 1 through 8 characters long. For Message Digest 5 (MD5) authentication, it can be 1 through 16 characters long. If you include spaces, enclose all characters in quotation marks (" ").

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring VRRP Authentication (IPv4 Only) on page 188
- Configuring VRRP Authentication (IPv4 Only)
- authentication-type on page 214
- version-3 on page 231

authentication-type

Syntax authentication-type *authentication*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Enable Virtual Router Redundancy Protocol (VRRP) IPv4 authentication and specify the authentication scheme for the VRRP group. If you enable authentication, you must specify a password by including the **authentication-key** statement.

All routers in the VRRP group must use the same authentication scheme and password.

NOTE: When VRRPv3 is enabled, the **authentication-type** and **authentication-key** statements cannot be configured for any VRRP groups.

Options *authentication*—Authentication scheme:

- **simple**—Use a simple password. The password is included in the transmitted packet, so this method of authentication is relatively insecure.
- **md5**—Use the MD5 algorithm to create an encoded checksum of the packet. The encoded checksum is included in the transmitted packet. The receiving routing platform uses the authentication key to verify the packet, discarding it if the digest does not match. This algorithm provides a more secure authentication scheme.

Default: none (no authentication is performed).

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring VRRP Authentication (IPv4 Only) on page 188
- Configuring VRRP Authentication (IPv4 Only)
- authentication-key on page 213
- version-3 on page 231

bandwidth-threshold

Syntax

bandwidth-threshold *bits-per-second* priority-cost *priority*;

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track interface *interface-name*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track interface *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track interface *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description

Specify the bandwidth threshold for Virtual Router Redundancy Protocol (VRRP) logical interface tracking.

Options

bits-per-second—Bandwidth threshold for the tracked interface. When the bandwidth of the tracked interface drops below the specified value, the VRRP group uses the bandwidth threshold priority cost value. You can include up to five bandwidth threshold statements for each interface you track.

Range: 1 through 10000000000000 bits per second

priority-cost *priority*—The value subtracted from the configured VRRP priority when the tracked interface or route is down to force a new master router election. The sum of all the costs for all interfaces or routes that are tracked must be less than or equal to the configured priority of the VRRP group.

Required Privilege Level

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

Related Documentation

- Configuring a Logical Interface to Be Tracked on page 195
- Configuring a Logical Interface to Be Tracked

delegate-processing (VRRP)

Syntax delegate-processing;

Hierarchy Level [edit protocols vrrp]

Release Information Statement introduced in Junos OS Release 9.6.

Description Configure the distributed ppmd process to send VRRP advertisements.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration

Related Documentation

- Enabling the Distributed Periodic Packet Management Process for VRRP on page 201

fast-interval

Syntax fast-interval *milliseconds*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Configure the interval, in milliseconds, between Virtual Router Redundancy Protocol (VRRP) advertisement packets.

All routers in the VRRP group must use the same advertisement interval.

Options *milliseconds*—Interval between advertisement packets.

Range: 10 through 40,950 milliseconds (range extended from 100–999 to 10–40,950 in Junos OS Release 12.2).

NOTE: When configuring VRRP for IPv4, if you have chosen not to enable VRRPv3, you cannot set a value less than 100 for **fast-interval**. Commit check fails if a value less than 100 is configured.

Default: 1 second

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring the Advertisement Interval for the VRRP Master Router on page 189
- Configuring the Advertisement Interval for the VRRP Master
- advertise-interval on page 211
- advertise-interval on page 211
- inet6-advertise-interval on page 220
- version-3 on page 231

global-advertisements-threshold

Syntax global-advertisements-threshold *advertisement-value*;

Hierarchy Level [edit protocols vrrp]

Release Information Statement introduced in Junos OS Release 12.2.

Description Configure the number of fast advertisements that can be missed by a backup router before the master router is declared as down.

NOTE:

- The advertisement value configured using the **global-advertisements-threshold** statement is applicable to all the Virtual Router Redundancy Protocol (VRRP) groups in the system.
- Setting the advertisement value of the **global-advertisements-threshold** configuration to 1 is not recommended for a scaled configuration with an aggressive advertisement interval. For example, if you have 1000 VRRP groups with an advertisement interval of 100 ms, then do not set the **global-advertisements-threshold** value to 1.
- Changing the advertisement value of the **global-advertisements-threshold** configuration during runtime can result in unpredictable behavior by the VRRP state machine. For example, momentary ownership change from the master router to the backup router and vice versa. Therefore, avoid changing the advertisement value of the **global-advertisements-threshold** statement during runtime.

Options *advertisement-value*—Number of VRRP advertisements missed before the master router is declared as down.

Range: 1 through 15

Default: 3

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Improving the Convergence Time for VRRP on page 179
- Configuring VRRP to Improve Convergence Time on page 202

hold-time (VRRP)

Syntax hold-time *seconds*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* preempt],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* preempt],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* preempt],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* preempt]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description In a Virtual Router Redundancy Protocol (VRRP) configuration, set the hold time before a higher-priority backup router preempts the master router.

Default VRRP preemption is not timed.

Options *seconds*—Hold-time period.

Range: 0 through 3600 seconds

Default: 0 seconds (VRRP preemption is not timed.)

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring a Backup Router to Preempt the Master Router on page 192
- Configuring VRRP Preemption and Hold Time

inet6-advertise-interval

Syntax inet6-advertise-interval *milliseconds*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

Release Information Statement introduced in Junos OS Release 8.4R2.

Description Configure the interval between Virtual Router Redundancy Protocol (VRRP) IPv6 advertisement packets.

All routers in the VRRP group must use the same advertisement interval.

NOTE: When VRRPv3 is enabled, the **inet6-advertise-interval** statement cannot be used to configure advertisement intervals. Instead, use the **fast-interval** statement to configure advertisement intervals.

Options *milliseconds*—Interval, in milliseconds, between advertisement packets.

Range: 100 to 40,000 milliseconds (ms)

Default: 1 second

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring the Advertisement Interval for the VRRP Master Router on page 189
- advertise-interval on page 211
- fast-interval on page 217
- version-3 on page 231

interface (VRRP Group)

Syntax

```
interface interface-name {
  bandwidth-threshold bits-per-second priority-cost priority;
  priority-cost priority;
}
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track]

Release Information

Statement introduced before Junos OS Release 7.4.
bandwidth-threshold statement added in Junos OS Release 8.1.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description

Enable logical interface tracking for a Virtual Router Redundancy Protocol (VRRP) group.

Options

interface-name—Interface to be tracked for this VRRP group.

Range: 1 through 10 interfaces

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring a Logical Interface to Be Tracked on page 195
- Configuring a Logical Interface to Be Tracked
- Junos Services Interfaces Configuration Release 11.2

no-accept-data

See accept-data

no-preempt

See preempt (VRRP)

preempt (VRRP)

Syntax

```
(preempt | no-preempt) {
  hold-time seconds;
}
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description

In a Virtual Router Redundancy Protocol (VRRP) configuration, determine whether or not a backup router can preempt a master router:

- **preempt**—Allow the master router to be preempted.

NOTE: **By default, a higher-priority backup router can preempt a lower-priority master router.**

- **no-preempt**—Prohibit the preemption of the master router. When **no-preempt** is configured, the backup router cannot preempt the master router even if the backup router has a higher priority.

The remaining statement is explained separately.

Default

By default the **preempt** statement is enabled, and a higher-priority backup router preempts a lower-priority master router even if the **preempt** statement is not explicitly configured.

Required Privilege Level

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

Related Documentation

- Configuring a Backup Router to Preempt the Master Router on page 192
- Configuring VRRP Preemption and Hold Time

priority (Protocols VRRP)

Syntax priority *priority*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Configure a Virtual Router Redundancy Protocol (VRRP) router's priority for becoming the master default router. The router with the highest priority within the group becomes the master.

Options *priority*—Router's priority for being elected to be the master router in the VRRP group. A larger value indicates a higher priority for being elected.

Range: 1 through 255

Default: 100 (for backup routers)

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Basic VRRP Support on page 186
- Configuring Basic VRRP Support

priority-cost (VRRP)

Syntax priority-cost *priority*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track interface *interface-name*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track interface *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track interface *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track interface *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.2 for ACX2000 Universal Access Routers.

Description Configure a Virtual Router Redundancy Protocol (VRRP) router's priority cost for becoming the master default router. The router with the highest priority within the group becomes the master.

Options *priority*—The value subtracted from the configured VRRP priority when the tracked interface or route is down to force a new master router election. The sum of all the costs for all interfaces or routes that are tracked must be less than or equal to the configured priority of the VRRP group.

Range: 1 through 254

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring a Logical Interface to Be Tracked on page 195
- Configuring a Logical Interface to Be Tracked

priority-hold-time

Syntax priority-hold-time *seconds*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Configure a Virtual Router Redundancy Protocol (VRRP) router's priority hold time to define the minimum length of time that must elapse between dynamic priority changes. If the dynamic priority changes because of a tracking event, the priority hold timer begins running. If another tracking event or manual configuration change occurs while the timer is running, the new dynamic priority update is postponed until the timer expires.

NOTE: When the track feature is configured, and if VRRP should pre-empt due to the tracking interface or route transition, any configured pre-empt hold time will be ignored. VRRP master will pre-empt according to the configuration of the priority-hold time.

Options *seconds*—Minimum length of time that must elapse between dynamic priority changes.

Range: 0through 3600 seconds

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring a Logical Interface to Be Tracked on page 195
- Configuring a Logical Interface to Be Tracked

route (Interfaces)

Syntax route *prefix* routing-instance *instance-name* priority-cost *priority*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id* track],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id* track]

Release Information Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS 11.3 for QFX Series.
Statement introduced in Junos OS 12.1 for EX Series switches.

Description Enable route tracking for a Virtual Router Redundancy Protocol (VRRP) group.

Options *prefix*—Route to be tracked for this VRRP group.

priority-cost ***priority***—The value subtracted from the configured VRRP priority when the tracked interface or route is down, forcing a new master router election. The sum of all the costs for all interfaces or routes that are tracked must be less than or equal to the configured priority of the VRRP group.

routing-instance ***instance-name***—Routing instance in which the route is to be tracked. If the route is in the default, or global, routing instance, the value for ***instance-name*** must be **default**.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring a Route to Be Tracked on page 197
- Configuring a Route to Be Tracked

skew-timer-disable

Syntax skew-timer-disable;

Hierarchy Level [edit protocols vrrp]

Release Information Statement introduced in Junos OS Release 12.2.

Description Disable the skew timer, thereby reducing the time required to transition from the backup state to the master state.

NOTE: **The skew-timer-disable statement is used when there is only one master router and one backup router in the network.**

Default By default, the skew timer is enabled for all the VRRP groups.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Improving the Convergence Time for VRRP on page 179
- Configuring VRRP to Improve Convergence Time on page 202

startup-silent-period

Syntax startup-silent-period *seconds*;

Hierarchy Level [edit protocols vrrp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Instruct the system to ignore the Master Down Event when an interface transitions from the down state to the up state. This statement is used to avoid incorrect error alarms caused by the delay or interruption of incoming Virtual Router Redundancy Protocol (VRRP) advertisement packets during the interface startup phase.

Options *seconds*—Number of seconds for the startup period.

Default: 4 seconds

Range: 1 through 2000 seconds

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring the Startup Period for VRRP Operations on page 185

- Configuring the Startup Period for VRRP Operations

traceoptions (Protocols VRRP)

Syntax

```
traceoptions {
  file filename <files number> <match regular-expression> <microsecond-stamp> <size size>
    <world-readable | no-world-readable>;
  flag flag;
  no-remote-trace;
}
```

Hierarchy Level [edit protocols vrrp]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define tracing operations for the Virtual Router Redundancy Protocol (VRRP) process.

To specify more than one tracing operation, include multiple **flag** statements.

By default, VRRP logs the error, dcd configuration, and routing socket events in a file in the directory **/var/log**.

Default If you do not include this statement, no VRRP-specific tracing operations are performed.

Options **file *filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. By default, VRRP tracing output is placed in the file **vrrpd**.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. When the maximum number is reached, the oldest trace file is overwritten.

Range: 0 through 4,294,967,296 files

Default: 3 files

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. These are the VRRP-specific tracing options:

- **all**—All VRRP tracing operations
- **database**—Database changes
- **general**—General events
- **interfaces**—Interface changes
- **normal**—Normal events
- **packets**—Packets sent and received

- **state**—State transitions
- **timer**—Timer events

match ***regular-expression***—(Optional) Refine the output to include only those lines that match the given regular expression.

microsecond-stamp—(Optional) Provide a timestamp with microsecond granularity.

no-world-readable—(Optional) Restrict users from reading the log file.

size ***size***—(Optional) Maximum size of each trace file, in kilobytes, megabytes, or gigabytes. When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the ***trace-file*** again reaches its maximum size, ***trace-file*.0.** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.
Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB
Range: 10 KB through the maximum file size supported on your routing platform
Default: 1 MB

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

world-readable—(Optional) Allow users to read the log file.

Required Privilege Level

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

Related Documentation

- Tracing VRRP Operations on page 199

track (VRRP)

Syntax

```
track {
  interface interface-name {
    bandwidth-threshold bits-per-second priority-cost priority;
    priority-cost priority;
  }
  priority-hold-time seconds;
  route prefix/prefix-length routing-instance instance-name priority-cost priority;
}
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

Release Information

Statement introduced before Junos OS Release 7.4.

priority-hold-time statement added in Junos OS Release 8.1.

route statement added in Junos OS Release 9.0.

Statement introduced in Junos OS 11.3 for the QFX Series.

Description

Enable logical interface tracking, route tracking, or both, for a Virtual Router Redundancy Protocol (VRRP) group.

Options

The remaining statements are described separately.

Required Privilege Level

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

Related Documentation

- Configuring a Logical Interface to Be Tracked on page 195
- Configuring a Route to Be Tracked on page 197
- Configuring a Logical Interface to Be Tracked
- Configuring a Route to Be Tracked

version-3

Syntax version-3;

Hierarchy Level [edit protocols vrrp]

Release Information Statement introduced in Junos OS Release 12.2.

Description Enable Virtual Router Redundancy Protocol version 3 (VRRPv3).

NOTE:

- Even though the **version-3** statement can be configured only at the **[edit protocols vrrp]** hierarchy level, VRRPv3 is enabled on all the configured logical systems as well.
- When enabling VRRPv3, you must ensure that VRRPv3 is enabled on all the VRRP routers in the network. This is because VRRPv3 does not interoperate with the previous versions of VRRP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Junos OS Support for VRRPv3 on page 176
- VRRP Configuration Hierarchy on page 183
- VRRP for IPv6 Configuration Hierarchy on page 184

virtual-address

Syntax virtual-address [*addresses*];

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Configure the addresses of the virtual routers in a Virtual Router Redundancy Protocol (VRRP) IPv4 or IPv6 group. You can configure up to eight addresses.

Options *addresses*—Addresses of one or more virtual routers. Do not include a prefix length. If the address is the same as the interface's physical address, the interface becomes the master virtual router for the group.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Basic VRRP Support on page 186
- Configuring Basic VRRP Support

virtual-inet6-address

Syntax virtual-inet6-address [*addresses*];

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the addresses of the virtual routers in a Virtual Router Redundancy Protocol (VRRP) IPv6 group. You can configure up to eight addresses.

Options *addresses*—Addresses of one or more virtual routers. Do not include a prefix length. If the address is the same as the interface's physical address, the interface becomes the master virtual router for the group.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Basic VRRP Support on page 186

virtual-link-local-address

Syntax virtual-link-local-address *ipv6-address*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

Release Information Statement introduced in Junos OS Release 8.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description Configure a virtual link-local address for a Virtual Router Redundancy Protocol (VRRP) IPv6 group. You must explicitly define a virtual link-local address for each VRRP for IPv6 group. The virtual link-local address must be in the same subnet as the physical interface address.

NOTE: You do *not* need to configure link-local addresses and virtual link-local addresses when configuring VRRP for IPv6. Junos OS automatically generates link-local addresses and virtual link-local addresses. However, if link local addresses and virtual link-local addresses are configured, Junos OS considers the configured addresses.

Options ***ipv6-address***—virtual link-local IPv6 address for VRRP for an IPv6 group.

Range: 0 through 255

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Basic VRRP Support on page 186
- Junos OS Support for VRRPv3 on page 176

vrrp-group

Syntax

```
vrrp-group group-id {
  (accept-data | no-accept-data);
  advertise-interval seconds;
  authentication-key key;
  authentication-type authentication;
  fast-interval milliseconds;
  (preempt | no-preempt) {
    hold-time seconds;
  }
  priority number;
  track {
    interface interface-name {
      bandwidth-threshold bits-per-second priority-cost priority;
      priority-cost priority;
    }
    priority-hold-time seconds;
    route prefix/prefix-length routing-instance instance-name priority-cost priority;
  }
  virtual-address [ addresses ];
  vrrp-inherit-from vrrp-group;
}
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS 11.3 for the QFX Series.

Description

Configure a Virtual Router Redundancy Protocol (VRRP) IPv4 group.

Options

group-id—VRRP group identifier. If you enable MAC source address filtering on the interface, you must include the virtual MAC address in the list of source MAC addresses that you specify in the **source-address-filter** statement. MAC addresses ranging from **00:00:5e:00:01:00** through **00:00:5e:00:01:ff** are reserved for VRRP, as defined in RFC 2338. The VRRP group number must be the decimal equivalent of the last hexadecimal byte of the virtual MAC address.

Range: 0 through 255

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring Basic VRRP Support on page 186
- Example: Configuring VRRP on page 203
- Configuring Basic VRRP Support
- Example: Configuring VRRP for Load Sharing

- vrrp-inet6-group on page 235

vrrp-inet6-group

Syntax

```
vrrp-inet6-group group-id {
  (accept-data | no-accept-data);
  fast-interval milliseconds;
  inet6-advertise-interval seconds;
  (preempt | no-preempt) {
    hold-time seconds;
  }
  priority number;
  track {
    interface interface-name {
      bandwidth-threshold bits-per-second priority-cost priority;
      priority-cost priority;
    }
    priority-hold-time seconds;
    route prefix/prefix-length routing-instance instance-name priority-cost priority;
  }
  virtual-inet6-address [ addresses ];
  virtual-link-local-address ipv6-address;
}
```

Hierarchy Level

[edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address*]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Configure a Virtual Router Redundancy Protocol (VRRP) IPv6 group.

Options

group-id—VRRP group identifier. If you enable MAC source address filtering on the interface, you must include the virtual MAC address in the list of source MAC addresses that you specify in the **source-address-filter** statement. MAC addresses ranging from **00:00:5e:00:01:00** through **00:00:5e:00:01:ff** are reserved for VRRP, as defined in RFC 2338. The VRRP group number must be the decimal equivalent of the last hexadecimal byte of the virtual MAC address.

Range: 0 through 255

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring Basic VRRP Support on page 186
- VRRP for IPv6 Configuration Hierarchy on page 184

PART 8

Unified ISSU

CHAPTER 20

Unified ISSU Overview

This chapter includes the following sections:

Unified ISSU Concepts

A unified in-service software upgrade (unified ISSU) enables you to upgrade between two different Junos OS Releases with no disruption on the control plane and with minimal disruption of traffic. Unified ISSU is only supported on dual Routing Engine platforms. In addition, the graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) must be enabled.

A unified ISSU provides the following benefits:

- Eliminates network downtime during software image upgrades
- Reduces operating costs, while delivering higher service levels
- Allows fast implementation of new features

NOTE: The master Routing Engine and backup Routing Engine must be running the same software version before you can perform a unified ISSU.

You cannot take any PICs online or offline during a unified ISSU.

NOTE: You can verify the unified ISSU-compatibility of the software, hardware, and the configuration on a device by issuing the **request system software validate in-service-upgrade** command. This command runs the validation checks, and shows whether the operating system, device components, and configurations are ISSU compatible or not. For more information about the **request system software validate in-service-upgrade** command, see Junos OS Operational Mode Commands.

NOTE: Unicast RPF-related statistics are not saved across a unified ISSU, and the unicast RPF counters are reset to zero during a unified ISSU.

To perform a unified ISSU, complete the following steps:

1. Enable graceful Routing Engine switchover and nonstop active routing. Verify that the Routing Engines and protocols are synchronized.
2. Download the new software package from the Juniper Networks Support website and then copy the package to the router.
3. Issue the **request system software in-service-upgrade** command on the master Routing Engine.

A Junos OS Release package comprises three distinct systems:

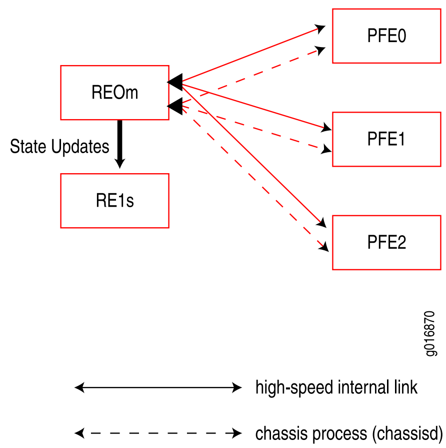
- Juniper Networks Operating System, which provides system control and all the features and functions of the Juniper Networks router that executes in the Routing Engines
- Juniper Networks Packet Forwarding Engine, which supports the high-performance traffic forwarding and packet handling capabilities
- Interface control

After the **request system software in-service-upgrade** command is issued, the following process occurs.

NOTE: In the illustrations, a solid line indicates the high-speed internal link between a Routing Engine and a Packet Forwarding Engine. A dotted line indicates the chassis process (chassisd), another method of communication between a Routing Engine and a Packet Forwarding Engine. RE0m and RE1s indicate master and backup (or standby) Routing Engines, respectively.

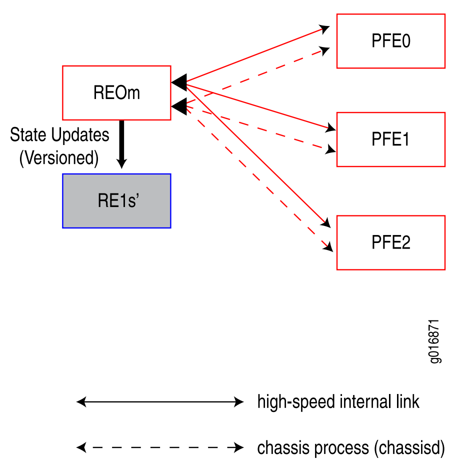
NOTE: The following process pertains to all supported routing platforms except the TX Matrix router. For information about the unified ISSU process on the TX Matrix router, see "Unified ISSU Process on the TX Matrix Router" on page 244. On M320 and T320 routers and on T640 and T1600 routers, the Packet Forwarding Engine resides on an FPC. However, on an M120 router, the Forwarding Engine Board (FEB) replaces the functions of a Packet Forwarding Engine. In the illustrations and steps, when considering an M120 router, you can regard the PFE as an FPC. As an additional step on an M120 router, after the FPCs and PICs have been upgraded, the FEBs are upgraded.

1. The master Routing Engine validates the router configuration to ensure that it can be committed using the new software version. Checks are made for disk space available for the **/var** file system on both Routing Engines, unsupported configurations, and for unsupported Physical Interface Cards (PICs). If there is not sufficient disk space available on either of the Routing Engines, the unified ISSU process fails and returns an error message saying that the Routing Engine does not have enough disk space available. However, unsupported PICs do not prevent a unified ISSU. The software issues a warning to indicate that these PICs will restart during the upgrade. Similarly, an unsupported protocol configuration does not prevent a unified ISSU. The software issues a warning that packet loss may occur for the protocol during the upgrade.



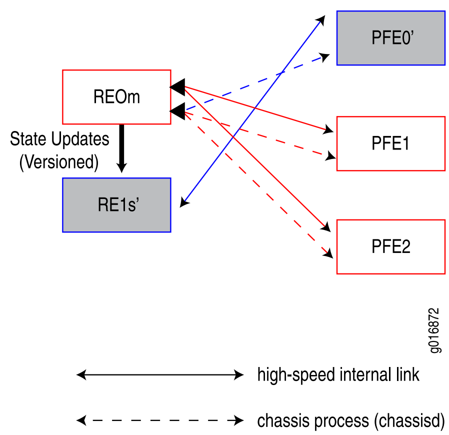
2. When the validation succeeds, the kernel state synchronization daemon (ksyncd) synchronizes the kernel on the backup Routing Engine with the master Routing Engine.

3. The backup Routing Engine is upgraded with the new software image. Before being upgraded, the backup Routing Engine gets the configuration file from the master Routing Engine and validates the configuration to ensure that it can be committed using the new software version. After being upgraded, it is resynchronized with the master Routing Engine. In the illustration, an apostrophe (') indicates the device is running the new version of software.



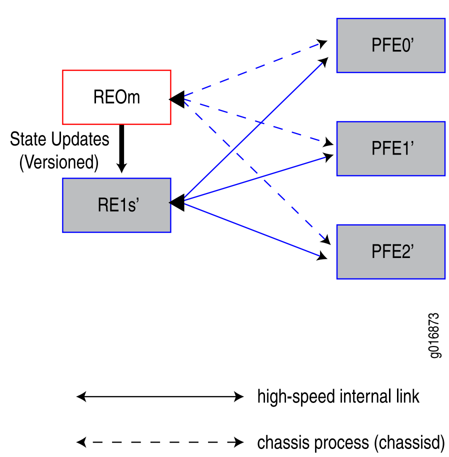
4. The chassis process (chassisd) on the master Routing Engine prepares other software processes for the unified ISSU. When all the processes are ready, chassisd sends an ISSU_PREPARE message to the Flexible PIC Concentrators (FPCs) installed in the router.

5. The Packet Forwarding Engine on each FPC saves its state and downloads the new software image from the backup Routing Engine. Next, each Packet Forwarding Engine sends an ISSU_READY message to the chassis process (chassisd).



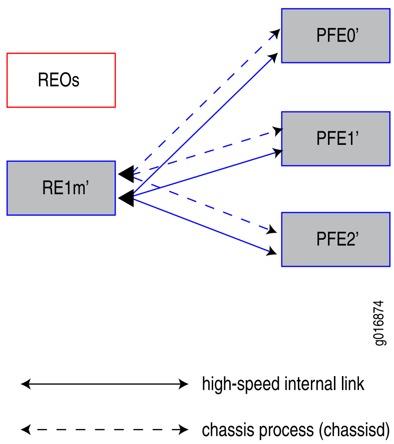
6. After receiving an ISSU_READY message from a Packet Forwarding Engine, the chassis process (chassisd) sends an ISSU_REBOOT message to the FPC on which the Packet Forwarding Engine resides. The FPC reboots with the new software image. After the FPC is rebooted, the Packet Forwarding Engine restores the FPC state and a high-speed internal link is established with the backup Routing Engine running the new software. The chassis process (chassisd) is also reestablished with the master Routing Engine.

7. After all Packet Forwarding Engines have sent a READY message using the chassis process (chassisd) on the master Routing Engine, other software processes are prepared for a Routing Engine switchover. The system is ready for a switchover at this point.

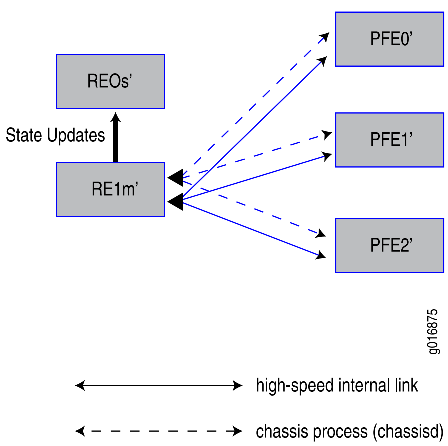


NOTE: In the case of an M120 router, the FEBs are upgraded at this point. When all FEBs have been upgraded, the system is ready for a switchover.

8. The Routing Engine switchover occurs and the backup Routing Engine becomes the new master Routing Engine.



9. The new backup Routing Engine is now upgraded to the new software image. (This step is skipped if the **no-old-master-upgrade** option is specified.)



10. When the backup Routing Engine has been successfully upgraded, the unified ISSU is complete.

Related Documentation

- Unified ISSU Process on the TX Matrix Router on page 244
- Unified ISSU System Requirements on page 245
- Best Practices on page 259
- Before You Begin on page 260
- Performing a Unified ISSU on page 263

Unified ISSU Process on the TX Matrix Router

After you issue the **request system software in-service-upgrade** command on a TX Matrix router, the following process occurs.

1. The management process (mgd) on the master Routing Engine of the TX Matrix router (global master) checks whether nonstop active routing (NSR) and graceful Routing Engine switchover (GRES) are enabled in the current configuration.
2. After successful validation of nonstop active routing and graceful Routing Engine switchover configuration, the management process copies the new image to the backup Routing Engines on the TX Matrix router and the T640 routers.
3. The kernel synchronization process (ksyncd) on the backup Routing Engines synchronizes the kernel on the backup Routing Engines with that of the master Routing Engines.
4. The backup Routing Engines are upgraded with the new software and are rebooted. After rebooting, the backup Routing Engines are once again synchronized with the global master Routing Engine.
5. The unified ISSU control moves from the management process to the chassis process (chassisd). The chassis process informs the software processes about the unified ISSU and waits for responses from various software processes (such as spmb).
6. After receiving messages from the software processes indicating that the processes are ready for unified ISSU, the chassis process on the global master Routing Engine sends messages to the chassis process on the routing nodes to start the unified ISSU.
7. The chassis process on the routing nodes sends ISSU_PREPARE messages to the field replaceable units (FRUs), such as FPC and intelligent PICs.
8. On receiving an ISSU_PREPARE message, the Packet Forwarding Engines save the current state information and download the new software image from the backup Routing Engines. Next, each Packet Forwarding Engine sends ISSU_READY messages to the chassis process.
9. On receiving an ISSU_READY message from the Packet Forwarding Engines, the chassis process sends an ISSU_REBOOT message to the FRUs. While the upgrade is in progress, the FRUs keep sending ISSU_IN_PROGRESS messages to the chassis process on the routing nodes. The chassis process on each routing node, in turn, sends an ISSU_IN_PROGRESS message to the chassis process on the global master Routing Engine.
10. After the reboot, the Packet Forwarding Engines restore the saved state information and connect back to the routing nodes; the chassis process on each routing node sends an ISSU_READY message to the chassis process on the global master Routing Engine. The ISSU_READY message from the chassis process on the routing nodes indicates that the unified ISSU is complete on the FRUs.
11. The unified ISSU control moves back to the management process on the global master Routing Engine.

12. The management process initiates Routing Engine switchover on the master Routing Engines.
13. Routing Engine switchover occurs on the TX Matrix router and the T640 routers.

NOTE: Currently, the FRUs on a TX Matrix router do not support graceful Routing Engine switchover and are rebooted every time graceful Routing Engine switchover occurs.

14. After the switchover, the FRUs connect to the new master Routing Engines, and the chassis manager and PFE manager on the T640 router FRUs connect to the new master Routing Engines on the T640 routers.
15. The management process on the global master Routing Engine initiates the upgrade process on the old master Routing Engines on the T640 routers.
16. After the old master Routing Engines on the T640 routers are upgraded, the management process initiates the upgrade of the old global master Routing Engine, that is, the old master Routing Engine on the TX Matrix router.
17. After a successful unified ISSU, the TX Matrix router and the T640 routers are rebooted.

Related Documentation

- Unified ISSU Concepts on page 239
- Unified ISSU System Requirements on page 245
- Best Practices on page 259
- Before You Begin on page 260
- Performing a Unified ISSU on page 263

Unified ISSU System Requirements

This section contains the following topics:

Unified ISSU Junos OS Release Support

In order to perform a unified ISSU, your router must be running a Junos OS Release that supports unified ISSU for the specific platform. See "Unified ISSU Platform Support" on page 246. You can use unified ISSU to upgrade from an ISSU-capable software release to a newer software release. However, note that:.

- The unified ISSU process is aborted if the Junos OS version specified for installation is a version earlier than the one currently running on the device. To downgrade from an ISSU-capable release to a previous software release (ISSU-capable or not), use the **request system add** command. Unlike an upgrade using the unified ISSU process, a downgrade using the **request system add** command can cause network disruptions and loss of data. For more information about the use of the **request system add** command, see the Installation and Upgrade Guide.
- The unified ISSU process is aborted if the specified upgrade has conflicts with the current configuration, components supported, and so forth.
- Unified ISSU does not support extension application packages developed using the Junos SDK.

Unified ISSU Platform Support

Table 10 on page 246 lists the platforms on which a unified ISSU is supported.

Table 10: Unified ISSU Platform Support

Routing Platform	Junos OS Release
M120 router	9.2 or later
M320 router	9.0 or later
M10i router with Enhanced Compact Forwarding Engine Board (CFEB-E)	9.5 or later
MX Series devices NOTE: Unified ISSU for MX Series routers does not support IEEE 802.1ag OAM and IEEE 802.3ah protocols.	9.3 or later 11.2 or later on MX Series 3D Universal Edge Routers (with Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces).
T320 router	9.0 or later
T640 router	9.0 or later
T1600 router	9.1 or later
TX Matrix router	9.3 or later

Unified ISSU Protocol Support

Unified ISSU is dependent on nonstop active routing. Table 11 on page 247 lists the protocols that are supported during a unified ISSU.

Table 11: Unified ISSU Protocol Support

Protocol	Junos OS Release
BGP, except for BGP VPN services	9.0 or later
DHCP access model (subscriber access)	11.2 or later
IS-IS	9.0 or later
LDP	9.0 or later
LDP-based virtual private LAN service (VPLS)	9.3 or later
Layer 2 circuits	9.2 or later
Layer 3 VPNs using LDP	9.2 or later
Link Aggregation Control Protocol (LACP) on MX Series routers	9.4 or later
OSPF/OSPFv3	9.0 or later
PPPoE access model (subscriber access)	11.4 or later
Protocol Independent Multicast (PIM)	9.3 or later
Routing Information Protocol (RIP)	9.1 or later

Unified ISSU Support for the Layer 2 Control Protocol Process

Unified ISSU supports the Layer 2 Control Protocol process (l2cpd) on MX Series 3D Universal Edge Routers. In a Layer 2 bridge environment, spanning tree protocols share information about port roles, bridge IDs, and root path costs between bridges using special data frames called Bridge Protocol Data Units (BPDUs). The transmission of BPDUs is controlled by the l2cpd process. Transmission of hello BPDUs is important in maintaining adjacencies on the peer systems.

The transmission of periodic packets on behalf of the l2cpd process is carried out by periodic packet management (PPM), which, by default, is configured to run on the Packet Forwarding Engine. The ppmd process on the Packet Forwarding Engine ensures that the BPDUs are transmitted even when the l2cpd process control plane is unavailable, and keeps the remote adjacencies alive during unified ISSU. However, if you want the distributed PPM (ppmd) process to run on the Routing Engine instead of the Packet Forwarding Engine, you can disable the ppmd process on the Packet Forwarding Engine, by including the **no-delegate-processing** statement at the [**edit routing-options ppm**] hierarchy level.

NOTE: The **delegate-processing** statement at the **[edit routing-options ppm]** hierarchy level, which was used to enable the ppmd process on the Packet Forwarding Engine in Junos OS Release 9.3 and earlier, has been deprecated as the ppmd process is enabled on the Packet Forwarding Engine by default in Junos OS Release 9.4 and later.

Unified ISSU enhancements and nonstop active bridging support for the l2cpd process ensure that the new master Routing Engine is able to take control during unified ISSU without any disruptions in the control plane and minimize the disruptions in the Layer 2 data plane during unified ISSU.

Unified ISSU Feature Support

Unified ISSU supports most Junos OS features in Junos OS Release 9.0. However, the following constraints apply:

- Link Aggregation Control Protocol (LACP)—Link changes are not processed until after the unified ISSU is complete.
- Automatic Protection Switching (APS)—Network changes are not processed until after the unified ISSU is complete.
- Ethernet Operation, Administration, and Management (OAM) as defined by IEEE 802.3ah and by IEEE 802.1ag—When a Routing Engine switchover occurs, the OAM hello times out, triggering protocol convergence.
- Ethernet circuit cross-connect (CCC) encapsulation—Circuit changes are not processed until after the unified ISSU is complete.
- Logical systems—On routers that have logical systems configured on them, only the master logical system supports unified ISSU.

Unified ISSU PIC Support

The following sections list the Physical Interface Cards (PICs) that are supported during a unified ISSU.

NOTE: For information about FPC types, FPC/PIC compatibility, and the initial Junos OS Release in which an FPC supported a particular PIC, see the PIC guide for your router platform.

PIC Considerations

Take the following PIC restrictions into consideration before performing a unified ISSU:

- Unsupported PICs—If a PIC is not supported by unified ISSU, at the beginning of the upgrade the software issues a warning that the PIC will be brought offline. After the PIC is brought offline and the ISSU is complete, the PIC is brought back online with the new firmware.
- PIC combinations—For some PICs, newer Junos OS services can require significant Internet Processor ASIC memory, and some configuration rules might limit certain combinations of PICs on particular platforms. With a unified ISSU:
  - If a PIC combination is not supported by the software version that the router is being upgraded from, the upgrade will be aborted.
  - If a PIC combination is not supported by the software version to which the router is being upgraded, the in-service software upgrade will abort, even if the PIC combination is supported by the software version from which the router is being upgraded.
- Interface statistics—Interface statistics might be incorrect because:
  - During bootup of the new microkernel on the Packet Forwarding Engine (PFE), host-bound traffic is not handled and might be dropped, causing packet loss.
  - During the hardware update of the Packet Forwarding Engine and its interfaces, traffic is halted and discarded. (The duration of the hardware update depends on the number and type of interfaces and on the router configuration.)
  - During a unified ISSU, periodic statistics collection is halted. If hardware counters saturate or wrap around, the software does not display accurate interface statistics.
- CIR oversubscription—If oversubscription of committed rate information (CIR) is configured on logical interfaces:
  - And the sum of the CIR exceeds the physical interface's bandwidth, after a unified in-service software upgrade is performed, each logical interface might not be given its original CIR.
  - And the sum of the delay buffer rate configured on logical interfaces exceeds the physical interface's bandwidth, after a unified in-service software upgrade is performed, each logical interface might not receive its original delay-buffer-rate calculation.

SONET/SDH PICs

Table 12 on page 250 lists the SONET/SDH PICs that are supported during a unified ISSU.

Table 12: Unified ISSU PIC Support: SONET/SDH

PIC Type	Number of Ports	Model Number	Router
OC3c/STM1	4-port	PB-4OC3-SON-MM—(EOL) PB-4OC3-SON-SMIR—(EOL)	M120 M320, T320, T640, T1600
		PE-4OC3-SON-MM—(EOL) PE-4OC3-SON-SMIR—(EOL)	M10i
	2-port	PE-2OC3-SON-MM—(EOL) PE-2OC3-SON-SMIR—(EOL)	
OC3c/STM1 with SFP	2-port	PE-2OC3-SON-SFP	M10i
OC3c/STM1, SFP (Multi-Rate)	4 OC3 ports, 4 OC12 ports	PB-4OC3-4OC12-SON-SFP	M120 M320, MX Series, T320, T640, T1600
	4 OC3 ports, 1 OC12 port	PB-4OC3-1OC12-SON-SFP PB-4OC3-1OC12-SON2-SFP	
		PE-4OC3-1OC12-SON-SFP	M10i
OC12c/STM4	1-port	PE-1OC12-SON-SFP PE-1OC12-SON-MM—(EOL) PE-1OC12-SON-SMIR—(EOL)	M10i
		PB-1OC12-SON-MM—(EOL) PB-1OC12-SON-SMIR—(EOL)	M120, M320, T320, T640, T1600, TX Matrix
	4-port	PB-4OC12-SON-MM PB-4OC12-SON-SMIR	
OC12c/STM4, SFP	1-port	PB-1OC12-SON-SFP	M120, M320, T320, T640, T1600, TX Matrix
OC48c/STM16, SFP	1-port	PB-1OC48-SON-SFP PB-1OC48-SON-B-SFP	M120, M320, MX Series, T320, T640, T1600, TX Matrix
	4-port	PC-4OC48-SON-SFP	
OC192/STM64	1-port	PC-1OC192-SON-VSR	MX Series routers

Table 12: Unified ISSU PIC Support: SONET/SDH *(continued)*

PIC Type	Number of Ports	Model Number	Router
OC192/STM64, XFP	1-port	PC-1OC192-SON-LR PC-1OC192-SON-SR2 PC-1OC192-VSR	M320, T320, T640, T1600
OC192/STM64, XFP	4-port	PD-4OC192-SON-XFP	M120, T640, T1600
	1-port	PC-1OC192-SON-XFP	MX Series routers
OC768/STM256	1-port	PD-1OC768-SON-SR	T640, T1600

Fast Ethernet and Gigabit Ethernet PICs

Table 13 on page 251 lists the Fast Ethernet and Gigabit Ethernet PICs that are supported during a unified ISSU.

NOTE: Starting with Junos OS Release 9.2, new Ethernet IQ2 PIC features might cause the software to reboot the PIC when a unified ISSU is performed. For information about applicable new Ethernet IQ2 PIC features, refer to the release notes for the specific Junos OS Release.

Table 13: Unified ISSU PIC Support: Fast Ethernet and Gigabit Ethernet

PIC Type	Number of Ports	Model Number	Router
Fast Ethernet	4	PB-4FE-TX	M120, M320, T320, T640, T1600, TX Matrix
		PE-4FE-TX	M10i
	8	PB-8FE-FX	M120, M320
		PE-8FE-FX	M10i
	12	PB-12FE-TX-MDI PB-12FE-TX-MDIX	M120, M320, T320
		PE-12FE-TX-MDI PE-12FE-TX-MDIX	M10i
	48	PB-48FE-TX-MDI PB-48FE-TX-MDIX	M120, M320, T320

Table 13: Unified ISSU PIC Support: Fast Ethernet and Gigabit Ethernet *(continued)*

PIC Type	Number of Ports	Model Number	Router
Gigabit Ethernet, SFP	1	PE-1GE-SFP	M10i
		PB-1GE-SFP	M120, M320, T320, T640, T1600, TX Matrix
	2	PB-2GE-SFP	
	4	PB-4GE-SFP	
	10	PC-10GE-SFP	
Gigabit Ethernet IQ, SFP	1	PE-1GE-SFP-QPP	M10i
		PB-1GE-SFP-QPP	M120, M320, T320, T640, T1600, TX Matrix
	2	PB-2GE-SFP-QPP	
Gigabit Ethernet IQ2, SFP	4	PB-4GE-TYPE1-SFP-IQ2	M120, M320, T320, T640, T1600, TX Matrix
	8	PB-8GE-TYPE2-SFP-IQ2 PC-8GE-TYPE3-SFP-IQ2	
Gigabit Ethernet IQ2, XFP	1	PC-1XGE-TYPE3-XFP-IQ2	M120, M320, T320, T640, T1600, TX Matrix
10-Gigabit Ethernet, XENPAK	1	PC-1XGE-XENPAK	M120, M320, T320, T640, T1600, TX Matrix
10-Gigabit Ethernet, DWDM	1	PC-1XGE-DWDM-CBAND	M120, M320, T320, T640, T1600, TX Matrix
10-Gigabit Ethernet	4	PD-4XGE-XFP NOTE: This PIC goes offline during a unified ISSU if the PIC is inserted on T-1600-FPC4-ES with revision number less than 13.	T640, T1600, TX Matrix, TX Matrix Plus
	10	PD-5-10XGE-SFPP	T640, T1600

Channelized PICs

Table 14 on page 253 lists the channelized PICs that are supported during a unified ISSU.

Table 14: Unified ISSU PIC Support: Channelized

PIC Type	Number of Ports	Model Number	Platform
Channelized E1 IQ	10	PB-10CHE1-RJ48-QPP	M120, M320, T320, T640, T1600, TX Matrix
		PE-10CHE1-RJ48-QPP-N	M10i
Channelized T1 IQ	10	PB-10CHT1-RJ48-QPP	M320, T320, T640, T1600
		PE-10CHT1-RJ48-QPP	M10i
Channelized OC IQ	1	PB-1CHOC12SMIR-QPP PB-1CHSTM1-SMIR-QPP PB-1CHOC3-SMIR-QPP	M120, M320, T320, T640, T1600, TX Matrix
		PE-1CHOC12SMIR-QPP PE-1CHOC3-SMIR-QPP	M10i
Channelized DS3 to DS0 IQ	4	PB-4CHDS3-QPP	M120, M320, T320, T640, T1600, TX Matrix
		PE-4CHDS3-QPP	M10i
Channelized STM 1	1	PE-1CHSTM1-SMIR-QPP	M10i

Tunnel Services PICs

Table 15 on page 253 lists the Tunnel Services PICs that are supported during a unified ISSU.

Table 15: Unified ISSU PIC Support: Tunnel Services

PIC Type	Model Number	Platform
1-Gbps Tunnel	PE-TUNNEL	M10i
	PB-TUNNEL-1	M120, M320, T320, T640, T1600, TX Matrix
4-Gbps Tunnel	PB-TUNNEL	
10-Gbps Tunnel	PC-TUNNEL	

ATM PICs

Table 16 on page 254 lists the ATM PICs that are supported during a unified ISSU. This includes support on Enhanced III FPCs.

Table 16: Unified ISSU PIC Support: ATM

PIC Type	Number of Ports	Model Number	Platform
DS3	4	PB-4DS3-ATM2	M120, M320, T320, T640, T1600, TX Matrix
		PE-4DS3-ATM2	M10i
E3	4	PB-4E3-ATM2	M120, M320, T320, T640, T1600, TX Matrix
	2	PE-2E3-ATM2	M10i
OC3/STM1	2	PB-2OC3-ATM2-MM PB-2OC3-ATM2-SMIR	M120, M320, T320, T640, T1600, TX Matrix
		PE-2OC3-ATM2-MM PE-2OC3-ATM2-SMIR	M10i
OC12/STM4	1	PB-1OC12-ATM2-MM PB-1OC12-ATM2-SMIR	M120, M320, T320, T640, T1600, TX Matrix
	2	PB-2OC12-ATM2-MM PB-2OC12-ATM2-SMIR	M120, M320, T320, T640, T1600, TX Matrix
	1	PE-1OC12-ATM2-MM PE-1OC12-ATM2-SMIR	M10i
OC48/STM16	1	PB-1OC48- ATM2-SFP	M120, M320, T320, T640, T1600, TX Matrix

Serial PICs

Unified ISSU supports the following 2-port EIA-530 serial PICs:

- PB-2EIA530 on M320 routers with Enhanced III FPCs, and on M120 routers
- PE-2EIA530 on M10i routers

DS3, E1, E3, and T1 PICs

Unified ISSU supports the following PICs on M120, M320, and T320 routers; T640 and T1600 routers; and the TX Matrix router:

- 4-Port DS3 PIC (PB-4DS3)
- 4-Port E1 Coaxial PIC (PB-4E1-COAX)
- 4-Port E1 RJ48 PIC (PB-4E1-RJ48)

- 4-port E3 IQ PIC (PB-4E3-QPP)
- 4-Port T1 PIC (PB-4T1-RJ48)

Unified ISSU supports the following PICs on M10i routers:

- 2-Port DS3 PIC (PE-2DS3)
- 4-Port DS3 PIC (PE-4DS3)
- 4-Port E1 PICs (PE-4E1-COAX and PE-4E1-RJ48)
- 2-Port E3 PIC (PE-2E3)
- 4-Port T1 PIC (PE-4T1-RJ48)
- 4-Port E3 IQ PIC (PE-4E3-QPP)

Enhanced IQ PICs

Unified ISSU supports the following PICs on M120, M320, and T320 routers; T640 and T1600 routers; and the TX Matrix router:

- 1-port Channelized OC12/STM4 enhanced IQ PIC (PB-1CHOC12-STM4-IQE-SFP)
- 1-port nonchannelized OC12/STM4 enhanced IQ PIC (PB-1OC12-STM4-IQE-SFP)
- 4-port Channelized DS3/E3 enhanced IQ PIC (PB-4CHDS3-E3-IQE-BNC)
- 4-port nonchannelized DS3/E3 enhanced IQ PIC (PB-4DS3-E3-IQE-BNC)
- 4-port nonchannelized SONET/SDH OC48/STM16 Enhanced IQ (IQE) PIC with SFP (PC-4OC48-STM16-IQE-SFP)

Unified ISSU supports 1-port Channelized OC48/STM16 Enhanced IQ (IQE) PIC with SFP (PB-1CHOC48-STM16-IQE-SFP) on MX Series routers.

Enhanced IQ2 Ethernet Services Engine (ESE) PIC

Unified ISSU supports the enhanced IQ2 ESE PICs listed in Table 17 on page 255.

Table 17: Unified ISSU Support: Enhanced IQ2 Ethernet Services Engine (ESE) PIC

Model Number	Number of Ports	Platform
PC-8GE-TYPE3-SFP-IQ2E	8	M120, M320, T320, T640, and TX Matrix.
PB-8GE-TYPE2-SFP-IQ2E	8	M120, M320, T320, T640, and TX Matrix.
PB-4GE-TYPE1-SFP-IQ2E	4	M120, M320, T320, and T640.
PC-1XGE-TYPE3-XFP-IQ2E	1	M120, M320, T320, T640, and TX Matrix.
PB-1CHOC48-STM16-IQE	1	M120, M320, T320, T640, and TX Matrix.
PE-4GE-TYPE1-SFP-IQ2E	4	M10i.

Table 17: Unified ISSU Support: Enhanced IQ2 Ethernet Services Engine (ESE) PIC *(continued)*

Model Number	Number of Ports	Platform
PE-4GE-TYPE1-SFP-IQ2	4	M10i.

Unified ISSU Support on MX Series 3D Universal Edge Routers

The following sections list the Dense Port Concentrators (DPCs), Flexible PIC Concentrators (FPCs), Modular Port Concentrators (MPCs), and Modular Interface Cards (MICs) that are supported during a unified ISSU on MX Series 3D Universal Edge Routers.

- Unified ISSU DPC and FPC Support on MX Series 3D Universal Edge Routers on page 256
- Unified ISSU MIC and MPC Support on MX Series 3D Universal Edge Routers on page 256
- Unified ISSU Limitation on MX Series 3D Universal Edge Routers on page 257

Unified ISSU DPC and FPC Support on MX Series 3D Universal Edge Routers

Unified ISSU supports all Dense Port Concentrators (DPCs) except the Multiservices DPC on the MX Series routers. However, unified ISSU does not support either of the FPCs (FPC type 2, **MX-FPC2**, or FPC type 3, **MX-FPC3**) on the MX Series routers. For more information about DPCs and FPCs on MX Series routers, go to http://www.juniper.net/techpubs/ en_US/release-independent/junos/ information-products/pathway-pages/mx-series/.

Unified ISSU MIC and MPC Support on MX Series 3D Universal Edge Routers

Unified ISSU supports all the Modular Port Concentrators (MPCs) and Modular Interface Cards (MICs) listed in Table 18 on page 256 and Table 19 on page 257. Unified ISSU is not supported on MX80 routers nor in an MX Series Virtual Chassis.

In the MPCs on MX Series routers, interface-specific and firewall filter statistics are preserved across a unified ISSU. During the unified ISSU, counter and policer operations are disabled.

To preserve statistics across a unified ISSU on MX Series routers with MPC/MIC interfaces, the router stores the statistics data as binary large objects. The router collects the statistics before the unified ISSU is initialized, and restores the statistics after the unified ISSU completes. No statistics are collected during the unified ISSU process.

To verify that statistics are preserved across the unified ISSU, you can issue CLI operational commands such as **show interfaces statistics** after the unified ISSU completes.

Table 18: Unified ISSU Support: MX Series 3D Universal Edge Routers

MPC Type	Number of Ports	Model Number	Platform
30-Gigabit Ethernet MPC	—	MX-MPC1-3D	MX Series 3D Universal Edge Routers
30-Gigabit Ethernet Queuing MPC	—	MX-MPC1-3D-Q	MX Series 3D Universal Edge Routers

Table 18: Unified ISSU Support: MX Series 3D Universal Edge Routers *(continued)*

MPC Type	Number of Ports	Model Number	Platform
60-Gigabit Ethernet MPC	—	MX-MPC2-3D	MX Series 3D Universal Edge Routers
60-Gigabit Ethernet Queuing MPC	—	MX-MPC2-3D-Q	MX Series 3D Universal Edge Routers
60-Gigabit Ethernet Enhanced Queuing MPC	—	MX-MPC2-3D-EQ	MX Series 3D Universal Edge Routers
10-Gigabit Ethernet MPC with SFP+	16	MPC-3D-16XGE-SFPP	MX Series 3D Universal Edge Routers

Table 19: Unified ISSU Support: MX Series 3D Universal Edge Routers

MIC Type	Number of Ports	Model Number	Platform
Gigabit Ethernet MIC with SFP	20	MIC-3D-20GE-SFP	MX Series 3D Universal Edge Routers
10-Gigabit Ethernet MICs with XFP	2	MIC-3D-2XGE-XFP	MX Series 3D Universal Edge Routers
10-Gigabit Ethernet MICs with XFP	4	MIC-3D-4XGE-XFP	MX Series 3D Universal Edge Routers
Tri-Rate Copper Ethernet MIC	40	MIC-3D-40GE-TX	MX Series 3D Universal Edge Routers

NOTE: Note that unified ISSU is supported only by the MICs listed in Table 19 on page 257.

Unified ISSU Limitation on MX Series 3D Universal Edge Routers

Unified in-service software upgrade (unified ISSU) is currently not supported when clock synchronization is configured for Synchronous Ethernet, Precision Time Protocol (PTP), and hybrid mode on MX80 3D Universal Edge Routers and on the MICs and MPCEs on MX240, MX480, and MX960 routers.

Related Documentation

- Unified ISSU Concepts on page 239
- Unified ISSU Process on the TX Matrix Router on page 244
- Before You Begin on page 260
- Performing a Unified ISSU on page 263

CHAPTER 21

Unified ISSU Configuration Guidelines

Best Practices

When you are planning to perform a unified in-service software upgrade (unified ISSU), choose a time when your network is as stable as possible. As with a normal upgrade, Telnet sessions, SNMP, and CLI access are briefly interrupted. In addition, the following restrictions apply:

- The master Routing Engine and backup Routing Engine must be running the same software version before you can perform a unified ISSU.
- During a unified ISSU, you cannot bring any PICs online or offline.
- Unicast RPF-related statistics are not saved across a unified ISSU, and the unicast RPF counters are reset to zero during a unified ISSU.

Related Documentation

- Before You Begin on page 260
- Performing a Unified ISSU on page 263
- Verifying a Unified ISSU on page 275
- Troubleshooting Unified ISSU Problems on page 276

Before You Begin

Before you begin a unified ISSU, complete the tasks in the following sections:

1. Verify That the Master and Backup Routing Engines Are Running the Same Software Version on page 261
2. Back Up the Router Software on page 261
3. Verify That Graceful Routing Engine Switchover and Nonstop Active Routing Are Configured on page 262

Verify That the Master and Backup Routing Engines Are Running the Same Software Version

To verify that both Routing Engines are running the same version of software, issue the following command:

```
{master}
user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: m320
JUNOS Base OS boot [9.0-20071210.0]
JUNOS Base OS Software Suite [9.0-20071210.0]
JUNOS Kernel Software Suite [9.0-20071210.0]
JUNOS Crypto Software Suite [9.0-20071210.0]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0-20071210.0]
JUNOS Packet Forwarding Engine Support (M320) [9.0-20071210.0]
JUNOS Online Documentation [9.0-20071210.0]
JUNOS Routing Software Suite [9.0-20071210.0]
re1:
--------------------------------------------------------------------------
Hostname: host
Model: m320
JUNOS Base OS boot [9.0-20071210.0]
JUNOS Base OS Software Suite [9.0-20071210.0]
JUNOS Kernel Software Suite [9.0-20071210.0]
JUNOS Crypto Software Suite [9.0-20071210.0]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0-20071210.0]
JUNOS Packet Forwarding Engine Support (M320) [9.0-20071210.0]
JUNOS Online Documentation [9.0-20071210.0]
JUNOS Routing Software Suite [9.0-20071210.0]
```

If both Routing Engines are not running the same software version, issue the **request system software add** command on the desired Routing Engine so that the software version is the same. For more information, see the Installation and Upgrade Guide.

Back Up the Router Software

As a preventive measure in case any problems occur during an upgrade, issue the **request system snapshot** command on *each* Routing Engine to back up the system software to the router's hard disk. The following is an example of issuing the command on the master Routing Engine:

```
{master}
user@host> request system snapshot
Verifying compatibility of destination media partitions...
Running newfs (220MB) on hard-disk media / partition (ad1s1a)...
Running newfs (24MB) on hard-disk media /config partition (ad1s1e)...
Copying '/dev/ad0s1a' to '/dev/ad1s1a' .. (this may take a few minutes)
Copying '/dev/ad0s1e' to '/dev/ad1s1e' .. (this may take a few minutes)
The following filesystems were archived: / /config
```

NOTE: The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. After you issue the **request system snapshot** command, the router's flash and hard disks are identical. You can return to the previous version of the software only by booting the router from removable media. For more information about the **request system snapshot** command, see the Junos OS System Basics Configuration Guide.

Verify That Graceful Routing Engine Switchover and Nonstop Active Routing Are Configured

Before you begin a unified ISSU, ensure that graceful Routing Engine switchover and nonstop active routing are configured on your router.

1. To verify graceful Routing Engine switchover is configured, on the backup Routing Engine (**re1**) issue the **show system switchover** command. The output should be similar to the following example. The **Graceful switchover** field state must be **On**.

```
{backup}
user@host> show system switchover
Graceful switchover: On
Configuration database: Ready
Kernel database: Ready
Peer state: Steady State
```

2. To verify nonstop active routing is configured, on the master Routing Engine (**re0**) issue the **show task replication** command. The output should be similar to the following example.

```
{master}

user@host> show task replication
        Stateful Replication: Enabled
        RE mode: Master

    Protocol                Synchronization Status
    OSPF                    Complete
    IS-IS                   Complete
```

If graceful Routing Engine switchover and nonstop active routing are not configured, complete the following steps:

1. On the master Routing Engine (**re0**), enable graceful Routing Engine switchover. Include the **graceful-switchover** statement at the **[edit chassis redundancy]** hierarchy level.
2. On the master Routing Engine, enable nonstop active routing. Include the **commit synchronize** statement at the **[edit system]** hierarchy level and the **nonstop-routing** statement at the **[edit routing-options]** hierarchy level.
3. On the master Router Engine, issue the **commit** command.

   The system provides the following confirmation that the master and backup Routing Engines are synchronized:

```
re0:
configuration check succeeds
```



```
re1:
commit complete
re0:
commit complete
```

Related Documentation

- Unified ISSU Concepts on page 239
- Unified ISSU Process on the TX Matrix Router on page 244
- Unified ISSU System Requirements on page 245
- Best Practices on page 259
- Performing a Unified ISSU on page 263
- Verifying a Unified ISSU on page 275
- Troubleshooting Unified ISSU Problems on page 276

Performing a Unified ISSU

You can perform a unified ISSU in one of three ways:

1. Upgrading and Rebooting Both Routing Engines Automatically on page 263
2. Upgrading Both Routing Engines and Rebooting the New Backup Routing Engine Manually on page 267
3. Upgrading and Rebooting Only One Routing Engine on page 272

Upgrading and Rebooting Both Routing Engines Automatically

When you issue the **request system software in-service-upgrade** command with the **reboot** option, the system automatically upgrades both Routing Engines to the newer software and reboots both Routing Engines. This option enables you to complete the unified ISSU with a single command.

To perform a unified ISSU using the **request system software in-service-upgrade *package-name* reboot** command, complete the following steps:

1. Download the software package from the Juniper Networks Support website, http://www.juniper.net/support/. Choose the Canada and U.S., Worldwide, or Junos-FIPS edition. Place the package on a local server. To download the package, you must have a service contract and an access account. If you do not have an access account, complete the registration form at the Juniper Networks website: https://www.juniper.net/registration/Register.jsp.
2. Copy the package to the router. We recommend that you copy it to the **/var/tmp** directory, which is a large file system on the hard disk.

   `user@host>`**file copy ftp://*username:prompt*@*ftp.hostname.net*/*filename*/var/tmp/*filename***
3. To verify the current software version running on both Routing Engines, on the master Routing Engine issue the **show version invoke-on all-routing-engines** command. The following example shows that both Routing Engines are running an image of Junos OS, Release 9.0, that was built on December 11, 2007:

```
{backup}
user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: m320
JUNOS Base OS boot [9.0-20071211.2]
JUNOS Base OS Software Suite 9.0-20071211.2]
JUNOS Kernel Software Suite [9.0-20071211.2]
JUNOS Crypto Software Suite [9.0-20071211.2]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0-20071211.2]
JUNOS Packet Forwarding Engine Support (M320) [9.0-20071211.2]
JUNOS Online Documentation [9.0-20071211.2]
JUNOS Routing Software Suite [9.0-20071211.2]

re1:
--------------------------------------------------------------------------
Hostname: host1
Model: m320
JUNOS Base OS boot [9.0-20071211.2]
JUNOS Base OS Software Suite [9.0-20071211.2]
JUNOS Kernel Software Suite [9.0-20071211.20]
JUNOS Crypto Software Suite [9.0-20071211.2]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0-20071211.2]
JUNOS Packet Forwarding Engine Support (M320) [9.0-20071211.2]
JUNOS Online Documentation [9.0-20071211.2]
JUNOS Routing Software Suite [9.0-20071211.2]
```

4. On the master Routing Engine, issue the **request system software in-service-upgrade *package-name* reboot** command. The following example upgrades the current version to an image of Junos OS, Release 9.0, that was built on January 14, 2008:

```
{master}
user@host> request system software in-service-upgrade
/var/tmp/jinstall-9.0-20080114.2-domestic-signed.tgz reboot
ISSU: Validating Image
PIC 0/3 will be offlined (In-Service-Upgrade not supported)
Do you want to continue with these actions being taken ? [yes,no] (no) yes

ISSU: Preparing Backup RE
Pushing bundle to re1
Checking compatibility with configuration
Initializing...
Using jbase-9.0-20080114.2
Verified manifest signed by PackageProduction_9_0_0
Using /var/tmp/jinstall-9.0-20080114.2-domestic-signed.tgz
Verified jinstall-9.0-20080114.2-domestic.tgz signed by PackageProduction_9_0_0
Using jinstall-9.0-20080114.2-domestic.tgz
Using jbundle-9.0-20080114.2-domestic.tgz
Checking jbundle requirements on /
Using jbase-9.0-20080114.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jkernel-9.0-20080114.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jcrypto-9.0-20080114.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jpfe-9.0-20080114.2.tgz
Using jdocs-9.0-20080114.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jroute-9.0-20080114.2.tgz
```



```
Verified manifest signed by PackageProduction_9_0_0
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
mgd: commit complete
Validation succeeded
Installing package '/var/tmp/jinstall-9.0-20080114.2-domestic-signed.tgz' ...
Verified jinstall-9.0-20080114.2-domestic.tgz signed by PackageProduction_9_0_0
Adding jinstall...
Verified manifest signed by PackageProduction_9_0_0

WARNING:     This package will load JUNOS 9.0-20080114.2 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving package file in /var/sw/pkg/jinstall-9.0-20080114.2-domestic-signed.tgz
 ...
Saving state for rollback ...
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU started
ISSU: Backup RE Prepare Done
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 6          Online (ISSU)
  FPC 7          Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
ISSU: Upgrading Old Master RE
Installing package '/var/tmp/paKEuy' ...
Verified jinstall-9.0-20080114.2-domestic.tgz signed by PackageProduction_9_0_0
Adding jinstall...
```

```
Verified manifest signed by PackageProduction_9_0_0

WARNING:     This package will load JUNOS 9.0-20080114.2 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving package file in /var/sw/pkg/jinstall-9.0-20080114.2-domestic-signed.tgz
 ...
cp: /var/tmp/paKEuy is a directory (not copied).
Saving state for rollback ...
ISSU: Old Master Upgrade Done
ISSU: IDLE
Shutdown NOW!
Reboot consistency check bypassed - jinstall 9.0-20080114.2 will complete
installation upon reboot
[pid 30227]

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

System going down IMMEDIATELY


Connection to host closed.
```

When the new backup (old master) Routing Engine is rebooted, you are logged out from the router.

5. After waiting a few minutes, log in to the router again. You are logged in to the new backup Routing Engine (**re0**). To verify that both Routing Engines have been upgraded, issue the following command:

```
{backup}
user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: m320
JUNOS Base OS boot [9.0-20080114.2]
JUNOS Base OS Software Suite 9.0-20080114.2]
JUNOS Kernel Software Suite [9.0-20080114.2]
JUNOS Crypto Software Suite [9.0-20080114.2]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0-20080114.2]
JUNOS Packet Forwarding Engine Support (M320) [9.0-20080114.2]
JUNOS Online Documentation [9.0-20080114.2]
JUNOS Routing Software Suite [9.0-20080114.2]
```



```
re1:
--------------------------------------------------------------------------
Hostname: host1
Model: m320
JUNOS Base OS boot [9.0-20080114.2]
JUNOS Base OS Software Suite [9.0-20080114.2]
JUNOS Kernel Software Suite [9.0-20080114.2]
JUNOS Crypto Software Suite [9.0-20080114.2]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0-20080114.2]
JUNOS Packet Forwarding Engine Support (M320) [9.0-20080114.2]
JUNOS Online Documentation [9.0-20080114.2]
JUNOS Routing Software Suite [9.0-20080114.2]
```

6. To make **re0** the master Routing Engine, issue the following command:

```
{backup}
user@host> request chassis routing-engine master acquire
Attempt to become the master routing engine ? [yes,no] (no) yes

Resolving mastership...
Complete. The local routing engine becomes the master.

{master}
user@host>
```

7. Issue the **request system snapshot** command on *each* Routing Engine to back up the system software to the router's hard disk.

NOTE: The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. After you issue the **request system snapshot** command, the router's flash and hard disks are identical. You can return to the previous version of the software only by booting the router from removable media.

Upgrading Both Routing Engines and Rebooting the New Backup Routing Engine Manually

When you issue the **request system software in-service-upgrade** command without any options, the system upgrades and reboots the new master Routing Engine to the newer software. The new software is placed on the new backup (old master) Routing Engine; however, to complete the upgrade, you must issue the **request system reboot** command on the new backup Routing Engine.

To perform a unified ISSU using the **request system software in-service-upgrade *package-name*** command without any options, complete the following steps:

1. Download the software package from the Juniper Networks Support website, http://www.juniper.net/support/. Choose the Canada and U.S., Worldwide, or Junos-FIPS edition. Place the package on a local server. To download the package, you must have a service contract and an access account. If you do not have an access account, complete the registration form at the Juniper Networks website: https://www.juniper.net/registration/Register.jsp.

2. Copy the package to the router. We recommend that you copy it to the **/var/tmp** directory, which is a large file system on the hard disk.

   `user@host>`**file copy ftp://*username:prompt*@*ftp.hostname.net*/*filename*/var/tmp/*filename***

3. To verify the current software version running on both Routing Engines, on the master Routing Engine, issue the **show version invoke-on all-routing-engines** command. The following example shows that both Routing Engines are running Junos OS Release 9.0R1:

```
{master}

user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: m320
JUNOS Base OS boot [9.0R1]
JUNOS Base OS Software Suite [9.0R1]
JUNOS Kernel Software Suite [9.0R1]
JUNOS Crypto Software Suite [9.0R1]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0R1]
JUNOS Packet Forwarding Engine Support (M320) [9.0R1]
JUNOS Online Documentation [9.0R1]
JUNOS Routing Software Suite [9.0R1]

re1:
--------------------------------------------------------------------------
Hostname: host1
Model: m320
JUNOS Base OS boot [9.0R1]
JUNOS Base OS Software Suite [9.0R1]
JUNOS Kernel Software Suite [9.0R1]
JUNOS Crypto Software Suite [9.0R1]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0R1]
JUNOS Packet Forwarding Engine Support (M320) [9.0R1]
JUNOS Online Documentation [9.0R1]
JUNOS Routing Software Suite [9.0R1]
```

4. On the master Routing Engine, issue the **request system software in-service-upgrade *package-name*** command. The following example upgrades the current version to Junos OS Release 9.0R1.2:

```
user@host> request system software in-service-upgrade
/var/tmp/jinstall-9.0R1.2-domestic-signed.tgz
ISSU: Validating Image
FPC 4 will be offlined (In-Service-Upgrade not supported)
Do you want to continue with these actions being taken ? [yes,no] (no) yes

ISSU: Preparing Backup RE
```



```
Pushing bundle to re1
Checking compatibility with configuration
Initializing...
Using jbase-9.0-20080117.0
Verified manifest signed by PackageProduction_9_0_0
Using /var/tmp/jinstall-9.0R1.2-domestic-signed.tgz
Verified jinstall-9.0R1.2-domestic.tgz signed by PackageProduction_9_0_0
Using jinstall-9.0R1.2-domestic.tgz
Using jbundle-9.0R1.2-domestic.tgz
Checking jbundle requirements on /
Using jbase-9.0R1.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jkernel-9.0R1.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jcrypto-9.0R1.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jpfe-9.0R1.2.tgz
Using jdocs-9.0R1.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jroute-9.0R1.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
mgd: commit complete
Validation succeeded
Installing package '/var/tmp/jinstall-9.0R1.2-domestic-signed.tgz' ...
Verified jinstall-9.0R1.2-domestic.tgz signed by PackageProduction_9_0_0
Adding jinstall...
Verified manifest signed by PackageProduction_9_0_0

WARNING:     This package will load JUNOS 9.0R1.2 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving package file in /var/sw/pkg/jinstall-9.0R1.2-domestic-signed.tgz ...
Saving state for rollback ...
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU started
ISSU: Backup RE Prepare Done
```

```
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 3          Online (ISSU)
  FPC 4          Offline                 Offlined by cli command
  FPC 5          Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
ISSU: Upgrading Old Master RE
Installing package '/var/tmp/paeBi5' ...
Verified jinstall-9.0R1.2-domestic.tgz signed by PackageProduction_9_0_0
Adding jinstall...
Verified manifest signed by PackageProduction_9_0_0

WARNING:     This package will load JUNOS 9.0R1.2 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving package file in /var/sw/pkg/jinstall-9.0R1.2-domestic-signed.tgz ...
cp: /var/tmp/paeBi5 is a directory (not copied).
Saving state for rollback ...
ISSU: Old Master Upgrade Done
ISSU: IDLE
```

5. Issue the **show version invoke-on all-routing-engines** command to verify that the new backup (old master) Routing Engine (**re0**), is still running the previous software image, while the new master Routing Engine (**re1**) is running the new software image:

```
{backup}

user@host> show version
re0:
--------------------------------------------------------------------------
Hostname: user
Model: m320
JUNOS Base OS boot [9.0R1]
JUNOS Base OS Software Suite [9.0R1]
JUNOS Kernel Software Suite [9.0R1]
JUNOS Crypto Software Suite [9.0R1]
```



```
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0R1]
JUNOS Packet Forwarding Engine Support (M320) [9.0R1]
JUNOS Online Documentation [9.0R1]
JUNOS Routing Software Suite [9.0R1]
labpkg [7.0]
JUNOS Installation Software [9.0R1.2]

re1:
--------------------------------------------------------------------------
Hostname: user1
Model: m320
JUNOS Base OS boot [9.0R1.2]
JUNOS Base OS Software Suite [9.0R1.2]
JUNOS Kernel Software Suite [9.0R1.2]
JUNOS Crypto Software Suite [9.0R1.2]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0R1.2]
JUNOS Packet Forwarding Engine Support (M320) [9.0R1.2]
JUNOS Online Documentation [9.0R1.2]
JUNOS Routing Software Suite [9.0R1.2]
```

6. At this point, if you choose not to install the newer software version on the new backup Routing Engine (**re1**), you can issue the **request system software delete jinstall** command on it. Otherwise, to complete the upgrade, go to the next step.

7. Reboot the new backup Routing Engine (**re0**) by issuing the **request system reboot** command:

```
{backup}

user@host> request system reboot
Reboot the system ? [yes,no] (no) yes

Shutdown NOW!
Reboot consistency check bypassed - jinstall 9.0R1.2 will complete installation
 upon reboot
[pid 6170]

{backup}
user@host>
System going down IMMEDIATELY


Connection to host closed by remote host.
Connection to host closed.
```

   If you are not on the console port, you are disconnected from the router session.

8. After waiting a few minutes, log in to the router again. You are logged in to the new backup Routing Engine (**re0**). To verify that both Routing Engines have been upgraded, issue the following command:

```
{backup}

user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: m320
JUNOS Base OS boot [9.0R1.2]
JUNOS Base OS Software Suite [9.0R1.2]
JUNOS Kernel Software Suite [9.0R1.2]
JUNOS Crypto Software Suite [9.0R1.2]
```



```
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0R1.2]
JUNOS Packet Forwarding Engine Support (M320) [9.0R1.2]
JUNOS Online Documentation [9.0R1.2]
JUNOS Routing Software Suite [9.0R1.2]

re1:
--------------------------------------------------------------------------
Hostname: host1
Model: m320
JUNOS Base OS boot [9.0R1.2]
JUNOS Base OS Software Suite [9.0R1.2]
JUNOS Kernel Software Suite [9.0R1.2]
JUNOS Crypto Software Suite [9.0R1.2]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.0R1.2]
JUNOS Packet Forwarding Engine Support (M320) [9.0R1.2]
JUNOS Online Documentation [9.0R1.2]
JUNOS Routing Software Suite [9.0R1.2]
```

9. To make **re0** the master Routing Engine, issue the following command:

```
{backup}
```

```
user@host> request chassis routing-engine master acquire
Attempt to become the master routing engine ? [yes,no] (no) yes

Resolving mastership...
Complete. The local routing engine becomes the master.

{master}
user@host>
```

10. Issue the **request system snapshot** command on *each* Routing Engine to back up the system software to the router's hard disk.

NOTE: The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. After you issue the **request system snapshot** command, the router's flash and hard disks are identical. You can return to the previous version of the software only by booting the router from removable media.

Upgrading and Rebooting Only One Routing Engine

When you issue the **request system software in-service-upgrade** command with the **no-old-master-upgrade** option, the system upgrades and reboots only the new master Routing Engine. To upgrade the new backup (former master) Routing Engine, you must issue the **request system software add** command.

To perform a unified ISSU using the **request system software in-service-upgrade** ***package-name*** **no-old-master-upgrade** commands, complete the following steps:

1. Download the software package from the Juniper Networks Support website, http://www.juniper.net/support/. Choose the Canada and U.S., Worldwide, or Junos-FIPS edition. Place the package on a local server. To download the package, you must have a service contract and an access account. If you do not have an access account, complete the registration form at the Juniper Networks website: https://www.juniper.net/registration/Register.jsp.

2. Copy the package to the router. We recommend that you copy it to the **/var/tmp** directory, which is a large file system on the hard disk.

   ```
   user@host>file copy ftp://username:prompt@ftp.hostname.net/filename/var/tmp/filename
   ```

3. To verify the current software version running on both Routing Engines, on the master Routing Engine issue the **show version invoke-on all-routing-engines** command. The following example shows that both Routing Engines are running an image of Junos OS Release 9.0 that was built on December 11, 2007:

   ```
   {backup}

   user@host> show version invoke-on all-routing-engines
   re0:
   --------------------------------------------------------------------------
   Hostname: host
   Model: m320
   JUNOS Base OS boot [9.0-20071211.2]
   JUNOS Base OS Software Suite 9.0-20071211.2]
   JUNOS Kernel Software Suite [9.0-20071211.2]
   JUNOS Crypto Software Suite [9.0-20071211.2]
   JUNOS Packet Forwarding Engine Support (M/T Common) [9.0-20071211.2]
   JUNOS Packet Forwarding Engine Support (M320) [9.0-20071211.2]
   JUNOS Online Documentation [9.0-20071211.2]
   JUNOS Routing Software Suite [9.0-20071211.2]

   re1:
   --------------------------------------------------------------------------
   Hostname: host1
   Model: m320
   JUNOS Base OS boot [9.0-20071211.2]
   JUNOS Base OS Software Suite [9.0-20071211.2]
   JUNOS Kernel Software Suite [9.0-20071211.20]
   JUNOS Crypto Software Suite [9.0-20071211.2]
   JUNOS Packet Forwarding Engine Support (M/T Common) [9.0-20071211.2]
   JUNOS Packet Forwarding Engine Support (M320) [9.0-20071211.2]
   JUNOS Online Documentation [9.0-20071211.2]
   JUNOS Routing Software Suite [9.0-20071211.2]
   ```

4. On the master Routing Engine, issue the **request system software in-service-upgrade** ***package-name*** **no-old-master-upgrade** command. The following example upgrades the current version to an image of Junos OS Release 9.0 that was built on January 16, 2008:

   ```
   {master}

   user@host> request system software in-service-upgrade
   /var/tmp/jinstall-9.0-20080116.2-domestic-signed.tgz no-old-master-upgrade
   ISSU: Validating Image
   ISSU: Preparing Backup RE
   ```

```
Pushing bundle to re1
Checking compatibility with configuration
Initializing...
Using jbase-9.0-20080116.2
Verified manifest signed by PackageProduction_9_0_0
Using /var/tmp/jinstall-9.0-20080116.2-domestic-signed.tgz
Verified jinstall-9.0-20080116.2-domestic.tgz signed by PackageProduction_9_0_0
Using jinstall-9.0-20080116.2-domestic.tgz
Using jbundle-9.0-20080116.2-domestic.tgz
Checking jbundle requirements on /
Using jbase-9.0-20080116.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jkernel-9.0-20080116.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jcrypto-9.0-20080116.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jpfe-9.0-20080116.2.tgz
Using jdocs-9.0-20080116.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Using jroute-9.0-20080116.2.tgz
Verified manifest signed by PackageProduction_9_0_0
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
mgd: commit complete
Validation succeeded
Installing package '/var/tmp/jinstall-9.0-20080116.2-domestic-signed.tgz' ...
Verified jinstall-9.0-20080116.2-domestic.tgz signed by PackageProduction_9_0_0
Adding jinstall...
Verified manifest signed by PackageProduction_9_0_0

WARNING:     This package will load JUNOS 9.0-20080116.2 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving package file in /var/sw/pkg/jinstall-9.0-20080116.2-domestic-signed.tgz
 ...
Saving state for rollback ...
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU started
```



```
ISSU: Backup RE Prepare Done
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 3          Online (ISSU)
  FPC 5          Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
Skipping Old Master Upgrade
ISSU: IDLE

{backup}
user@host>
```

5. You are now logged in to the new backup (old master Routing Engine). If you want to install the new software version on the new backup Routing Engine, issue the **request system software add /var/tmp/jinstall-9.0-20080116.2-domestic-signed.tgz** command.

Related Documentation

- Unified ISSU System Requirements on page 245
- Best Practices on page 259
- Before You Begin on page 260
- Verifying a Unified ISSU on page 275
- Troubleshooting Unified ISSU Problems on page 276
- Managing and Tracing BFD Sessions During Unified ISSU Procedures on page 276

Verifying a Unified ISSU

To verify the status of FPCs and their corresponding PICs after the most recent unified ISSU, issue the **show chassis in-service-upgrade** command on the master Routing Engine:

```
user@host> show chassis in-service-upgrade
  Item           Status                  Reason
  FPC 0          Online
  FPC 1          Online
  FPC 2          Online
    PIC 0        Online
    PIC 1        Online
  FPC 3          Offline                 Offlined by CLI command
  FPC 4          Online
    PIC 1        Online
  FPC 5          Online
    PIC 0        Online
  FPC 6          Online
```

```
  PIC 3         Online
FPC 7           Online
```

For more information about the **show chassis in-service-upgrade** command, see the Junos OS Operational Mode Commands.

Related Documentation

- Performing a Unified ISSU on page 263
- Troubleshooting Unified ISSU Problems on page 276
- Managing and Tracing BFD Sessions During Unified ISSU Procedures on page 276

Troubleshooting Unified ISSU Problems

If the unified ISSU procedure stops progressing, complete the following steps:

1. Open a new session on the master Routing Engine and issue the **request system software abort in-service-upgrade** command.
2. Check the existing router session to verify that the upgrade has been aborted.

   An “ISSU: aborted!” message is provided. Additional system messages provide you with information about where the upgrade stopped and recommendations for the next step to take.

For more information about the **request system software abort in-service-upgrade** command, see the Junos OS Operational Mode Commands.

Related Documentation

- Unified ISSU Concepts on page 239
- Unified ISSU Process on the TX Matrix Router on page 244
- Unified ISSU System Requirements on page 245
- Best Practices on page 259
- Performing a Unified ISSU on page 263
- Verifying a Unified ISSU on page 275
- Managing and Tracing BFD Sessions During Unified ISSU Procedures on page 276

Managing and Tracing BFD Sessions During Unified ISSU Procedures

Bidirectional Forwarding Detection (BFD) sessions temporarily increase their detection and transmission timers during unified ISSU procedures. After the upgrade, these timers revert to the values in use before the unified ISSU started. The BFD process replicates the unified ISSU state and timer values to the backup Routing Engine for each session.

No additional configuration is necessary to enable unified ISSU for BFD. However, you can disable the BFD timer negotiation during the unified ISSU by including the **no-issu-timer-negotiation** statement at the **[edit protocols bfd]** hierarchy level:

```
[edit protocols bfd]
no-issu-timer-negotiation;
```

If you configure this statement, the BFD timers maintain their original values during unified ISSU.

CAUTION: **The sessions might flap during unified ISSU or Routing Engine switchover, depending on the detection intervals.**

For more information about BFD, see the Junos OS Routing Protocols Configuration Guide.

To configure unified ISSU trace options for BFD sessions, include the **issu** statement at the **[edit protocols bfd traceoptions flag]** hierarchy level.

```
[edit protocols]
bfd {
  traceoptions {
    flag issu;
  }
}
```

Related Documentation

- Unified ISSU Concepts on page 239
- Unified ISSU Process on the TX Matrix Router on page 244
- Unified ISSU System Requirements on page 245
- Best Practices on page 259
- Before You Begin on page 260
- Performing a Unified ISSU on page 263
- Verifying a Unified ISSU on page 275
- Troubleshooting Unified ISSU Problems on page 276

CHAPTER 22

Unified ISSU Configuration Statements Summary

This chapter provides a reference for each of the unified in-service software upgrade (ISSU) configuration statements. The statements are organized alphabetically.

NOTE: To perform a unified ISSU, you must first configure graceful Routing Engine switchover and nonstop active routing (NSR).

no-issu-timer-negotiation

Syntax no-issu-timer-negotiation;

Hierarchy Level [edit protocols bfd],
[edit logical-systems *logical-system-name* protocols bfd],
[edit routing-instances *routing-instance-name* protocols bfd]

Release Information Statement introduced in Junos OS Release 9.1.

Description Disable unified ISSU timer negotiation for Bidirectional Forwarding Detection (BFD) sessions.

CAUTION: The sessions might flap during unified ISSU or Routing Engine switchover, depending on the detection intervals.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Managing and Tracing BFD Sessions During Unified ISSU Procedures on page 276
- Junos OS Routing Protocols Configuration Guide.

traceoptions (Protocols BFD)

Syntax

```
traceoptions {
  file name <size size> <files number> <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level [edit protocols bfd]

Release Information Statement introduced before Junos OS Release 7.4.
issu flag for BFD added in Junos OS Release 9.1.

Description Define tracing operations that track unified in-service software upgrade (ISSU) functionality in the router.

To specify more than one tracing operation, include multiple **flag** statements.

Default If you do not include this statement, no global tracing operations are performed.

Options **disable**—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *name*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place global routing protocol tracing output in the file **routing-log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

Range: 2 through 1000 files

Default: 2 files

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

flag *flag*—Tracing operation to perform. There is only one unified ISSU tracing option:

- **issu**—Trace BFD unified ISSU operations.

no-world-readable—Restrict users from reading the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 128 KB

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

world-readable—Allow users to read the log file.

Required Privilege Level

routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Managing and Tracing BFD Sessions During Unified ISSU Procedures on page 276

PART 9

Interchassis Redundancy for MX Series Routers Using Virtual Chassis

CHAPTER 23

MX Series Interchassis Redundancy Overview

Interchassis Redundancy and Virtual Chassis Overview

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

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

Interchassis Redundancy Overview

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

Interchassis redundancy is a high availability feature that prevents network outages and protects routers against access link failures, uplink failures, and wholesale chassis failures without visibly disrupting the attached subscribers or increasing the network management burden for service providers. Network outages can cause service providers to lose revenues and require them to register formal reports with government agencies. A robust interchassis redundancy implementation enables service providers to fulfill strict service-level agreements (SLAs) and avoid unplanned network outages to better meet the needs of their customers.

Virtual Chassis Overview

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

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

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

Supported Platforms for MX Series Virtual Chassis

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

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

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

Benefits of Configuring a Virtual Chassis

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

- Simplifies network management of two routers that are either colocated or geographically dispersed across a Layer 2 point-to-point network.
- Provides resiliency against network outages and protects member routers against access link failures, uplink failures, and chassis failures without visibly disrupting attached subscribers or increasing the network management burden for service providers.
- Extends the high availability capabilities of applications such as graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) beyond a single MX Series router to both member routers in the Virtual Chassis.
- Enables service providers to fulfill strict service level agreements (SLAs) and avoid unplanned network outages to better meet their customers' needs.
- Provides the ability to scale bandwidth and service capacity as more high-priority voice and video traffic is carried on the network.

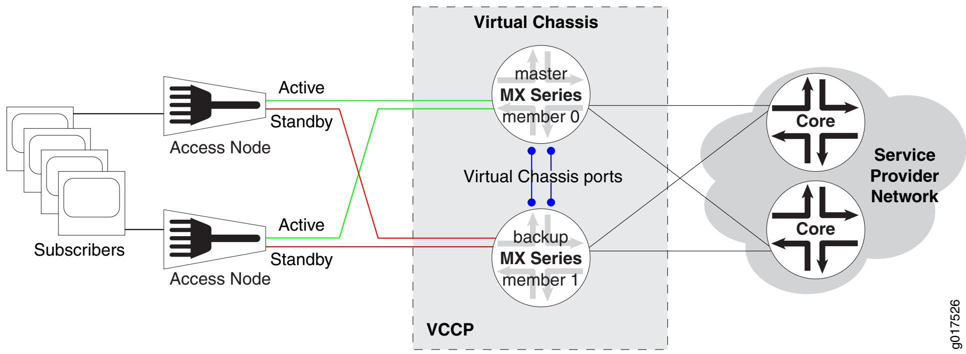
Related Documentation

- Virtual Chassis Components Overview on page 288
- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Virtual Chassis Components Overview

A Virtual Chassis configuration for MX Series 3D Universal Edge Routers interconnects two MX Series routers into a logical system that you can manage as a single network element. Figure 9 on page 288 illustrates a typical topology for a two-member MX Series Virtual Chassis.

Figure 9: Sample Topology for MX Series Virtual Chassis



This overview describes the basic hardware and software components of the Virtual Chassis configuration illustrated in Figure 9 on page 288, and covers the following topics:

Virtual Chassis Master Router

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

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

- Manages both the master and backup member routers
- Runs the chassis management processes and control protocols

- Receives and processes all incoming and exception path traffic destined for the Virtual Chassis
- Propagates the Virtual Chassis configuration (including member IDs, roles, and configuration group definitions and applications) to the members of the Virtual Chassis

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

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

Virtual Chassis Backup Router

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

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

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

Virtual Chassis Line-card Router

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

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

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

Virtual Chassis Ports

Virtual Chassis ports are special Ethernet interfaces that form a point-to-point connection between the member routers in a Virtual Chassis. When you create a Virtual Chassis, you must configure the Virtual Chassis ports on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces. After you configure a Virtual Chassis port, it is renamed **vcp-*slot/pic/port*** (for example, **vcp-2/2/0**), and the line card associated with that port comes online. For example, the sample Virtual Chassis topology shown in Figure 9 on page 288 has a total of four Virtual Chassis ports (represented by the blue dots), two on each of the two member routers.

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

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

Virtual Chassis port interfaces carry both VCCP packets and internal control and data traffic. Because the internal control traffic is neither encrypted nor authenticated, make sure the Virtual Chassis port interfaces are properly secured to prevent malicious third-party attacks on the data.

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

Virtual Chassis Port Trunks

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

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

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

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

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

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

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

NOTE: For information about how the slot numbering in an MX Series Virtual Chassis affects the use of SNMP, see "Virtual Chassis Slot Number Mapping for Use with SNMP" on page 399.

Virtual Chassis Control Protocol

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

- Discovers and builds the Virtual Chassis topology

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

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

Member IDs, Roles, and Serial Numbers

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

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

The preprovisioned configuration permanently associates the member ID and role with the member router's chassis serial number. When a new member router joins the Virtual Chassis, the VCCP software compares the router's serial number against the values specified in the preprovisioned configuration. If the serial number of a joining router does not match any of the configured serial numbers, the VCCP software prevents that router from becoming a member of the Virtual Chassis.

Related Documentation

- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Guidelines for Configuring Virtual Chassis Ports on page 293
- Global Roles and Local Roles in a Virtual Chassis on page 294
- Split Detection Behavior in a Virtual Chassis on page 301
- Virtual Chassis Slot Number Mapping for Use with SNMP on page 399
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Guidelines for Configuring Virtual Chassis Ports

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

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

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

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

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

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

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

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

  For a two-member MX Series Virtual Chassis, we recommend that you configure a minimum of two 10-Gigabit Ethernet Virtual Chassis ports on different line cards in each member router, for a total minimum of four 10-Gigabit Ethernet Virtual Chassis ports in the Virtual Chassis. In addition, make sure the Virtual Chassis port bandwidth is equivalent to no less than 50 percent of the aggregate bandwidth required for user data traffic. The following examples illustrate these recommendations:

  - If the bandwidth in your network is equivalent to two 10-Gigabit Ethernet interfaces (20 Gbps) on the access-facing side of the Virtual Chassis and two 10-Gigabit Ethernet interfaces (20 Gbps) on the core-facing side of the Virtual Chassis, we recommend that you configure two 10-Gigabit Ethernet Virtual Chassis ports, which is the recommended minimum in a Virtual Chassis for redundancy purposes.

  - If the aggregate bandwidth in your network is equivalent to ten 10-Gigabit Ethernet interfaces (100 Gbps), we recommend that you configure a minimum of five

10-Gigabit Ethernet Virtual Chassis ports, which is 50 percent of the aggregate bandwidth.

- A user data packet traversing the Virtual Chassis port interfaces between member routers is discarded at the Virtual Chassis egress port if the MTU size of the packet exceeds 9150 bytes.

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

Related Documentation

- Virtual Chassis Components Overview on page 288
- Configuring Virtual Chassis Ports to Interconnect Member Routers on page 326
- Class of Service Overview for Virtual Chassis Ports on page 305
- Guidelines for Configuring Class of Service for Virtual Chassis Ports on page 310
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Global Roles and Local Roles in a Virtual Chassis

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

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

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

- Role Name Format on page 295
- Global Role and Local Role Descriptions on page 295

Role Name Format

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

VC-*GlobalRole*<*LocalRole*>

where:

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

Global Role and Local Role Descriptions

Table 20 on page 295 describes the global roles and local roles in an MX Series Virtual Chassis.

Table 20: Global Roles and Local Roles in an MX Series Virtual Chassis

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

Table 20: Global Roles and Local Roles in an MX Series Virtual Chassis *(continued)*

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

Related Documentation

- Virtual Chassis Components Overview on page 288
- Mastership Election in a Virtual Chassis on page 297
- Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 332
- Disabling Split Detection in a Virtual Chassis Configuration on page 335

Mastership Election in a Virtual Chassis

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

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

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

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

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

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

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

   ```
   {master:member1-re0}
   user@host> show virtual-chassis protocol database member 0 extensive
    ...
     TLVs:
       Node Info: Member ID: 1, VC ID: 5a6a.e747.8511, Flags: 3, Priority: 129
          System ID: 001d.b510.0800, Device ID: 1
    ...
   ```

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

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

   - **129**—The **routing-engine** role is assigned to the member router.
   - **128**—No role is assigned to the member router.

- **0**—The **line-card** role is assigned to the member router (not supported in the current release).

  To display the mastership priority value for the member routers in the Virtual Chassis, issue the **show virtual-chassis status** command.

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

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

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

Related Documentation

- Virtual Chassis Components Overview on page 288
- Global Roles and Local Roles in a Virtual Chassis on page 294
- Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 332

Switchover Behavior in a Virtual Chassis

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

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

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

- Virtual Chassis Role Transitions During a Global Switchover on page 299
- Virtual Chassis Role Transitions During a Local Switchover on page 300
- GRES Readiness in a Virtual Chassis Configuration on page 301

Virtual Chassis Role Transitions During a Global Switchover

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

A global switchover in an MX Series Virtual Chassis causes the role transitions listed in Table 21 on page 299.

Table 21: Virtual Chassis Role Transitions During Global Switchover

Virtual Chassis Role *Before* Global Switchover	Virtual Chassis Role *After* Global Switchover
Virtual Chassis master router (VC-M)	Virtual Chassis backup router (VC-B)
Virtual Chassis backup router (VC-B)	Virtual Chassis master router (VC-M)
Master Routing Engine in the Virtual Chassis master router (VC-Mm)	Master Routing Engine in the Virtual Chassis backup router (VC-Bm)
Standby Routing Engine in the Virtual Chassis master router (VC-Ms)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)

Table 21: Virtual Chassis Role Transitions During Global Switchover *(continued)*

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

The local roles (**master** and **standby**, or **m** and **s**) of the Routing Engines do not change after a global switchover. For example, as shown in Table 21 on page 299, the master Routing Engine in the Virtual Chassis backup router (VC-Bm) remains the master Routing Engine in the Virtual Chassis master router (VC-Mm) after the global switchover.

Virtual Chassis Role Transitions During a Local Switchover

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

A local switchover in an MX Series Virtual Chassis causes the role transitions listed in Table 22 on page 300.

Table 22: Virtual Chassis Role Transitions During Local Switchover

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

The local roles (**master** and **standby**, or **m** and **s**) of the Routing Engines in the Virtual Chassis master router change after a local switchover, but the local roles of the Routing Engines in the Virtual Chassis backup router do not change. For example, as shown in Table 22 on page 300, the master Routing Engine in the Virtual Chassis master router (VC-Mm) becomes the standby Routing Engine in the Virtual Chassis backup router (VC-Bs) after the local switchover. By contrast, the master Routing Engine in the Virtual

Chassis backup router (VC-Bm) remains the master Routing Engine in the Virtual Chassis master router (VC-Mm) after the local switchover.

GRES Readiness in a Virtual Chassis Configuration

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

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

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

Related Documentation

- Virtual Chassis Components Overview on page 288
- Global Roles and Local Roles in a Virtual Chassis on page 294
- Mastership Election in a Virtual Chassis on page 297
- Switching the Global Master and Backup Roles in a Virtual Chassis Configuration on page 332
- Determining GRES Readiness in a Virtual Chassis Configuration on page 334
- Understanding Graceful Routing Engine Switchover in the Junos OS on page 51

Split Detection Behavior in a Virtual Chassis

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

This topic covers:

How Split Detection Works in a Virtual Chassis

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

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

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

Effect of Split Detection on Virtual Chassis Failure Scenarios

The behavior of a Virtual Chassis during certain failure scenarios depends on whether split detection is enabled or disabled. Table 23 on page 303 describes the effect of the split detection setting on common failure scenarios in a two-member MX Series Virtual Chassis.

Table 23: Effect of Split Detection on Common Virtual Chassis Failure Scenarios

Type of Failure	Split Detection Setting	Results
Virtual Chassis port interfaces go down	Enabled	• VC-B takes VC-M role. • Previous VC-M takes **line-card** (VC-L) role. The **line-card** role isolates the router and removes it from the Virtual Chassis until connectivity is restored. • Result is a single-member Virtual Chassis consisting of only a single VC-M. The VC-M continues to maintain subscriber state information and route traffic. When Virtual Chassis port interfaces are reconnected: • VC-M retains VC-M role. • VC-L takes VC-B role. • Subscribers are not affected.
Virtual Chassis port interfaces go down	Disabled	When Virtual Chassis port interfaces are disconnected: • VC-M retains VC-M role, and VC-B also takes VC-M role. The result is a Virtual Chassis with two VC-M routers, each of which maintains subscriber state information. • Initially, both VC-M routers have a complete list of subscribers. Because the two routers have the same configuration, the effect on subscribers, traffic patterns, behavior of external applications, and subscriber login and logout operations is unpredictable while the Virtual Chassis port interfaces are disconnected. When Virtual Chassis port interfaces are reconnected: • Original VC-M before the disconnection resumes VC-M role, and original VC-B before the disconnection resumes VC-B role. • Subscribers on the VC-M are preserved. • Subscribers on the VC-B are purged. • The subscribers preserved on the VC-M are unaffected, and all remaining subscribers are able to log back in to the router.

Table 23: Effect of Split Detection on Common Virtual Chassis Failure Scenarios *(continued)*

Type of Failure	Split Detection Setting	Results
Virtual Chassis backup router (VC-B) goes down	Enabled	• VC-M takes **line-card** (VC-L) role, which causes all line cards (FPCs) that do not host Virtual Chassis ports to go offline. • Previous VC-B is out of service. • The **line-card** role isolates the master router and removes it from the Virtual Chassis until connectivity is restored. As a result, the Virtual Chassis is left without a master router, which halts interchassis routing and effectively disables the Virtual Chassis configuration. When the failed router is brought back into service: • The mastership election algorithm is run to determine whether the router takes a VC-M or VC-B role. The Virtual Chassis then becomes operational. • All subscribers can log back in to the router. • Previous subscriber state information is not preserved.
Virtual Chassis backup router (VC-B) goes down	Disabled	• VC-M retains VC-M role and maintains all Virtual Chassis ports. • Previous VC-B is out of service. • Result is a single-member Virtual Chassis consisting of only a single VC-M. The VC-M continues to maintain subscriber state information and route traffic.
Virtual Chassis master router (VC-M) goes down	Split detection setting has no effect on behavior	• VC-B takes over VC-M role regardless of whether split detection is enabled or disabled. • Previous VC-M is out of service. • Result is a single-member Virtual Chassis consisting of only a single VC-M. The new VC-M continues to maintain subscriber state information and route traffic. When the original VC-M is brought back into service, or when the original VC-M is replaced with a new router: • Original VC-M or its replacement takes VC-B role. • Subscribers are not affected.

Table 23: Effect of Split Detection on Common Virtual Chassis Failure Scenarios *(continued)*

Type of Failure	Split Detection Setting	Results
Active access link between the VC-M and the access node, such as a digital subscriber line access multiplexer (DSLAM), goes down	Split detection setting has no effect on behavior	• Previous standby access link becomes the active access link between the VC-B and the access node. • Traffic is routed through the new active access link. • The VC-M continues to maintain subscriber state information and route traffic.

Related Documentation

- Virtual Chassis Components Overview on page 288
- Global Roles and Local Roles in a Virtual Chassis on page 294
- Mastership Election in a Virtual Chassis on page 297
- Switchover Behavior in a Virtual Chassis on page 299
- Disabling Split Detection in a Virtual Chassis Configuration on page 335

Class of Service Overview for Virtual Chassis Ports

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

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

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

Default CoS Configuration for Virtual Chassis Ports

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

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

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

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

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

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

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

Supported Platforms and Maximums for CoS Configuration of Virtual Chassis Ports

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

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

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

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

Default Classifiers for Virtual Chassis Ports

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

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

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

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

```
{master:member0-re0}

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



```
110                 network-control                     low
111                 network-control                     high
```

Default Rewrite Rules for Virtual Chassis Ports

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

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

```
{master:member0-re0}

user@host> show class-of-service rewrite-rule type ieee-802.1
Rewrite rule: ieee8021p-default, Code point type: ieee-802.1, Index: 34
  Forwarding class                    Loss priority       Code point
  best-effort                         low                 000
  best-effort                         high                001
  expedited-forwarding                low                 010
  expedited-forwarding                high                011
  assured-forwarding                  low                 100
  assured-forwarding                  high                101
  network-control                     low                 110
  network-control                     high                111
```

Default Scheduler Map for Virtual Chassis Ports

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

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

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

```
{master:member0-re0}

user@host> show class-of-service scheduler-map
Scheduler map: <default>, Index: 2

  Scheduler: <default-be>, Forwarding class: best-effort, Index: 21
    Transmit rate: 95 percent, Rate Limit: none, Buffer size: 95 percent, Buffer
 Limit: none, Priority: low
    Excess Priority: low
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             any             1    <default-drop-profile>
      Medium low      any             1    <default-drop-profile>
```



```
      Medium high     any             1     <default-drop-profile>
      High            any             1     <default-drop-profile>

  Scheduler: <default-nc>, Forwarding class: network-control, Index: 23
    Transmit rate: 5 percent, Rate Limit: none, Buffer size: 5 percent, Buffer
Limit: none, Priority: low
    Excess Priority: low
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             any             1    <default-drop-profile>
      Medium low      any             1    <default-drop-profile>
      Medium high     any             1    <default-drop-profile>
      High            any             1    <default-drop-profile>
  ...
```

Customized CoS Configuration for Virtual Chassis Ports

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

Output Traffic-Control Profiles

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

For an example that shows how to configure an output traffic-control profile customized for Virtual Chassis ports, see "Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers" on page 386.

Classifiers and Rewrite Rules

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

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

Per-Priority Shaping

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

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

Related Documentation

- Guidelines for Configuring Class of Service for Virtual Chassis Ports on page 310
- Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers on page 386

- Junos OS Class of Service Configuration Guide

Guidelines for Configuring Class of Service for Virtual Chassis Ports

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

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

Related Documentation

- Class of Service Overview for Virtual Chassis Ports on page 305

- Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers on page 386
- Junos OS Class of Service Configuration Guide

CHAPTER 24

Configuring MX Series Interchassis Redundancy Using Virtual Chassis

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

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

To configure a Virtual Chassis for MX Series routers:

1. Prepare your site for the Virtual Chassis configuration.

   See “Preparing for a Virtual Chassis Configuration” on page 315.

2. Install Junos OS licenses on the routers to be configured as members of the Virtual Chassis.

   See “Installing Junos OS Licenses on Virtual Chassis Member Routers” on page 317.

3. Define configuration groups for the Virtual Chassis.

   See “Creating and Applying Configuration Groups for a Virtual Chassis” on page 319.

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

   See “Configuring Preprovisioned Member Information for a Virtual Chassis” on page 321.

5. Enable graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) on both member routers.

   See “Enabling Graceful Routing Engine Switchover and Nonstop Active Routing for a Virtual Chassis” on page 323.

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

   See “Configuring Member IDs for a Virtual Chassis” on page 324.

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

   See “Configuring Virtual Chassis Ports to Interconnect Member Routers” on page 326.

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

   See the following topics:

   - Verifying the Status of Virtual Chassis Member Routers on page 402
   - Verifying the Operation of Virtual Chassis Ports on page 402
   - Verifying Neighbor Reachability for Member Routers in a Virtual Chassis on page 403
   - Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis on page 403
   - Viewing Information in the Virtual Chassis Control Protocol Adjacency Database on page 404
   - Viewing Information in the Virtual Chassis Control Protocol Link-State Database on page 404

- Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database on page 405
- Viewing Virtual Chassis Control Protocol Routing Tables on page 405
- Viewing Virtual Chassis Control Protocol Statistics for Member Routers and Virtual Chassis Ports on page 406

Related Documentation

- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Virtual Chassis Components Overview on page 288
- Guidelines for Configuring Virtual Chassis Ports on page 293
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Preparing for a Virtual Chassis Configuration

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

To prepare for configuring an MX Series Virtual Chassis:

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

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

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

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

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

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

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

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

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

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

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

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

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

7. Determine the desired location of the dedicated Virtual Chassis ports on both member routers, and use the Virtual Chassis ports to physically interconnect the member routers in a point-to-point topology.

8. Ensure that both MX Series routers to be configured as a member of the Virtual Chassis are running the same Junos OS release, and have basic network connectivity.

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Guidelines for Configuring Virtual Chassis Ports on page 293
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Installing Junos OS Licenses on Virtual Chassis Member Routers

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

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

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

Before you begin:

- Prepare your site for the Virtual Chassis configuration.

  See “Preparing for a Virtual Chassis Configuration” on page 315.
- Familiarize yourself with the procedures for installing and managing Junos OS licenses.

  See Installation and Upgrade Guide.

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

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

For example:

```
user@host> show system license
License usage:
                                 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
```

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

- Software Features That Require Licenses on MX Series Routers Only
- Installation and Upgrade Guide

Creating and Applying Configuration Groups for a Virtual Chassis

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

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

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

Using configuration group names of the form **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 to assign no IP address or an incorrect IP address to the **fxp0** management Ethernet interface, and could result in a display of the **Amnesiac** prompt during login.

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

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

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

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

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

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

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

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

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


user@gladius# **delete groups re1**

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

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

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

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

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

7. Commit the configuration.

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

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

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

- For more information about creating and managing configuration groups, see the CLI User Guide

Configuring Preprovisioned Member Information for a Virtual Chassis

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

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

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

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

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

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

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

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

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

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

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

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

   For example:

   [edit virtual-chassis]

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

5. Commit the configuration.

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

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

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

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

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

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

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

To enable graceful Routing Engine switchover and nonstop active routing:

1. Enable graceful Routing Engine switchover and nonstop active routing on member 0 (**gladius**):

   a. Log in to the console on member 0.

   b. Enable graceful switchover.

   ```
   [edit chassis redundancy]
   user@gladius# set graceful-switchover
   ```

   c. Enable nonstop active routing.

   ```
   [edit routing-options]
   user@gladius# set nonstop-routing
   ```

   d. Commit the configuration on member 0.

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

2. Enable graceful Routing Engine switchover and nonstop active routing on member 1 (**trefoil**):

   a. Log in to the console on member 1.

   b. Enable graceful switchover.

   ```
   [edit chassis redundancy]
   user@trefoil# set graceful-switchover
   ```

   c. Enable nonstop active routing.

   ```
   [edit routing-options]
   user@trefoil# set nonstop-routing
   ```

   d. Commit the configuration on member 1.

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


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

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

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345
- Configuring Graceful Routing Engine Switchover on page 59
- Configuring Nonstop Active Routing on page 91

Configuring Member IDs for a Virtual Chassis

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

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

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

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

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

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

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

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

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

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

For example:

```
{master:member0-re0}

user@hostA> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: 4f2b.1aa0.de08
                                              Mastership            Neighbor
List
Member ID       Status    Serial No    Model    priority    Role      ID
Interface

0 (FPC   0- 11) Prsnt    JN10C7135AFC mx240         129   Master*
```

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

   For example:

```
Amnesiac (ttyd0)

login: user
Password:
...

{master:member1-re0}

user> show virtual-chassis status
Virtual Chassis ID: ef98.2c6c.f7f7
                                              Mastership            Neighbor
List
Member ID       Status    Serial No    Model    priority    Role      ID
Interface

1 (FPC  12- 23) Prsnt    JN115D117AFB mx480         128   Master*
```

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

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Configuring Virtual Chassis Ports to Interconnect Member Routers

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

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

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

1. Configure the Virtual Chassis ports on the router configured as member 0.
   a. Configure the first Virtual Chassis port that connects to member 1.

      `{local:member0-re0}`

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

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

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

      `{local:member0-re0}`

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

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

      `{local:member0-re0}`

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

2. Configure the Virtual Chassis ports on the router configured as member 1.
   a. Configure the first Virtual Chassis port that connects to member 0.

      `{master:member1-re0}`

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

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

      `{master:member1-re0}`

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

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

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

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

3. (Optional) Verify 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 master router.

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

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

Related Documentation

- Guidelines for Configuring Virtual Chassis Ports on page 293
- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

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

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

To delete a Virtual Chassis configuration for MX Series routers:

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

   See "Deleting Virtual Chassis Ports in a Virtual Chassis Configuration" on page 329.

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

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

3. Delete the preprovisioned member information configured at the **[edit virtual-chassis]** hierarchy level on the master router.

4. Delete any interfaces that were configured on the member routers when the Virtual Chassis was created.

5. Delete the Virtual Chassis member IDs to reboot each router and disable Virtual Chassis mode.

   See "Deleting Member IDs in a Virtual Chassis Configuration" on page 330.

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

Related Documentation

- Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 360
- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Virtual Chassis Components Overview on page 288

Deleting Virtual Chassis Ports in a Virtual Chassis Configuration

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

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

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

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

   ```
   {master:member0-re0}
   user@host1> request virtual-chassis vc-port delete fpc-slot fpc-slot-number pic-slot pic-slot-number port port-number
   ```

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

   ```
   {master:member0-re0}
   user@host1> request virtual-chassis vc-port delete fpc-slot 2 pic-slot 2 port 0
   vc-port successfully deleted
   ```

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

   ```
   {master:member1-re0}
   user@host2> request virtual-chassis vc-port delete fpc-slot fpc-slot-number pic-slot pic-slot-number port port-number
   ```

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

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

   For member 0 (**host1**):

   ```
   {master:member0-re0}
   user@host1> show virtual-chassis vc-port all-members
   member0:
   --------------------------------------------------------------------------
   ```

   For member 1 (**host2**):

```
{backup:member1-re0}

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

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

Related Documentation

- Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 328
- Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 360
- Guidelines for Configuring Virtual Chassis Ports on page 293

Deleting Member IDs in a Virtual Chassis Configuration

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

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

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

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

```
{master:member0-re0}

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

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

{master:member0-re0}
user@host1>

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


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

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

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

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

{master:member1-re0}
user@host2>

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

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

   For example:

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

Related Documentation

- Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 328
- Example: Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 360

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

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

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

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

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

Related Documentation

- Example: Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 372
- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Virtual Chassis Components Overview on page 288

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

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

Before you begin:

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

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

  ```
  Error: mastership switch request NOT honored, backup not ready
  ```

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

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

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

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

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

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

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

  ```
  error: Virtual Chassis member is not the protocol master
  ```

  Issuing the **request virtual-chassis routing-engine master switch** command from the Virtual Chassis master router causes the global role transitions listed in Table 24 on page 333.

Table 24: Virtual Chassis Global Role Transitions Before and After Mastership Switchover

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

The local roles (**master** and **standby**, or **m** and **s**) of the Routing Engines do not change after you issue the **request virtual-chassis routing-engine master switch** command. For example, as shown in Table 24 on page 333, the master Routing Engine in the Virtual Chassis master router (VC-Mm) remains the master Routing Engine in the Virtual Chassis backup router (VC-Bm) after the switchover.

Related Documentation

- Virtual Chassis Components Overview on page 288
- Global Roles and Local Roles in a Virtual Chassis on page 294
- Mastership Election in a Virtual Chassis on page 297
- Switchover Behavior in a Virtual Chassis on page 299

Determining GRES Readiness in a Virtual Chassis Configuration

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

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

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

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

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

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

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

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

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

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

     ```
     {master:member0-re0}
     user@host> request virtual-chassis routing-engine master switch check
     error: chassisd Not ready for mastership switch, try after 217 secs.
     mastership switch request NOT honored, backup not ready
     ```

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

Related Documentation

- Switchover Behavior in a Virtual Chassis on page 299
- Virtual Chassis Components Overview on page 288
- Global Roles and Local Roles in a Virtual Chassis on page 294
- Understanding Graceful Routing Engine Switchover in the Junos OS on page 51

Disabling Split Detection in a Virtual Chassis Configuration

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

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

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

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

To disable split detection in an MX Series Virtual Chassis:

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

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

2. Commit the configuration.

Disabling split detection causes different results for different types of Virtual Chassis failures. For information, see "Split Detection Behavior in a Virtual Chassis" on page 301.

Related Documentation

- Split Detection Behavior in a Virtual Chassis on page 301
- Global Roles and Local Roles in a Virtual Chassis on page 294

- Switchover Behavior in a Virtual Chassis on page 299
- Virtual Chassis Components Overview on page 288

Accessing the Virtual Chassis Through the Management Interface

The management Ethernet interface (**fxp0**) on an MX Series router is an out-of-band management interface, also referred to as a management port, that enables you to use Telnet or SSH to access and manage the router remotely. You typically configure the management interface with an IP address and prefix length when you first install Junos OS.

You can configure a management Ethernet interface in one of two ways to access an MX Series Virtual Chassis:

- To access the Virtual Chassis as a whole, configure a consistent IP address for the management interface using the **master-only** option. You can use this management IP address to consistently access the master (primary) Routing Engine in the master router (protocol master) for the Virtual Chassis.
- To access a specific Routing Engine in an individual member router of the Virtual Chassis, configure an IP address for one of the following MX Series Virtual Chassis configuration groups:
  - **member0-re0**
  - **member0-re1**
  - **member1-re0**
  - **member1-re1**

BEST PRACTICE: **For most management tasks, we recommend that you access the Virtual Chassis as a whole through a consistent management IP address. For troubleshooting purposes, however, accessing a specific Routing Engine in an individual member router may be useful.**

To access an MX Series Virtual Chassis through the management Ethernet interface, do one of the following:

- Configure a consistent management IP address that accesses the entire Virtual Chassis through the master Routing Engine in the Virtual Chassis master router.

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

  For example, to access the entire Virtual Chassis via management IP address **10.4.5.33/16**:

  ```
  {master:member0-re0}[edit]
  user@host# set interfaces fxp0 unit 0 family inet address 10.4.5.33/16 master-only
  ```


- Configure a management IP address that accesses a specified Routing Engine in an individual member router in the Virtual Chassis.

  {master:member0-re0}[edit groups]
  user@host# **set member*n*-re*n* interfaces fxp0 unit 0 family inet address *ip-address/prefix-length***

  For example, to access the Routing Engine installed in slot 1 of member router 1 (**member1-re1**) in the Virtual Chassis:

  {master:member0-re0}[edit groups]
  user@host# **set member1-re1 interfaces fxp0 unit 0 family inet address 10.4.3.145/32**

Related Documentation

- Configuring a Consistent Management IP Address in the Junos® OS Network Interfaces

Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers

The Junos OS trace feature tracks Virtual Chassis operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems.

By default, tracing is disabled. When you enable the tracing operation on the router to be configured as the master (also referred to as the *protocol master*) of an MX Series Virtual Chassis, the default tracing behavior is as follows:

1. Important events are logged in a file with the name you specify in the **/var/log** directory. You cannot change the directory (**/var/log**) in which trace files are located.
2. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

   You can optionally specify the maximum number of trace files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB). (For more information about how log files are created, see the *Junos OS System Log Messages Reference*.)

   By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

To configure tracing of MX Series Virtual Chassis operations:

1. Configure a filename for the trace log.

   See "Configuring the Name of the Virtual Chassis Trace Log File" on page 338.
2. (Optional) Configure characteristics of the trace log file.

   See "Configuring Characteristics of the Virtual Chassis Trace Log File" on page 339.
3. (Optional) Configure user access to the trace log file.

   See "Configuring Access to the Virtual Chassis Trace Log File" on page 340.
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 341.

5. Configure flags to specify the Virtual Chassis operations that you want to trace.

   See “Configuring the Virtual Chassis Operations to Trace” on page 342.

Related Documentation

- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Configuring the Name of the Virtual Chassis Trace Log File

To trace operations for an MX Series Virtual Chassis, you must configure the name of the trace log file that the software saves in the **/var/log** directory.

To configure the filename for tracing MX Series Virtual Chassis operations:

- On the router to be designated as the master of the Virtual Chassis, specify the name of the trace log file.

```
[edit virtual-chassis]
user@host# set traceoptions file filename
```

Related Documentation

- Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 337
- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Configuring Characteristics of the Virtual Chassis Trace Log File

You can optionally configure the following characteristics of the trace log file for an MX Series Virtual Chassis:

- Maximum number of trace files—When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. You can optionally specify the maximum number of trace files to be from 2 through 1000. If you specify a maximum number of files with the **files** option, you must also specify a maximum file size with the **size** option.
- Maximum trace file size—You can configure the maximum trace file size to be from 10 KB through 1 gigabyte (GB) If you specify a maximum file size with the **size** option, you must also specify a maximum number of files with the **files** option.
- Timestamp—By default, timestamp information is placed at the beginning of each line of trace output. You can optionally prevent placement of a timestamp on any trace log file.
- Appending or replacing the trace file—By default, the router appends new information to an existing trace file. You can optionally specify that the router replace an exisiting trace file instead of appending information to it.

To configure the maximum number and maximum size of trace files:

- On the router to be designated as the master of the MX Series Virtual Chassis, specify the maximum number and maximum size of the trace file.

  [edit virtual-chassis]
  user@host# **set traceoptions file *filename* files *number* size *maximum-file-size***

  For example, to set the maximum number of files to 20 and the maximum file size to 2 MB for a trace file named **vccp**:

  [edit virtual-chassis]
  user@host# **set traceoptions file vccp files 20 size 2097152**

  When the **vccp** trace file for this example reaches 2 MB, **vccp** is renamed **vccp.0**, and a new file named **vccp** is created. When the new **vccp** file reaches 2 MB, **vccp.0** is renamed **vccp.1** and **vccp** is renamed **vccp.0**. This process repeats until there are 20 trace files. Then the oldest file (**vccp.19**) is overwritten by the newest file (**vccp.0**).

To prevent the router from the placing a timestamp on the trace log file:

- On the router to be designated as the master of the MX Series Virtual Chassis, specify that a timestamp not appear on the trace log file:

  [edit virtual-chassis]
  user@host# **set traceoptions file *filename* no-stamp**

To replace an exisiting trace file instead of appending information to it:

- On the router to be designated as the master of the MX Series Virtual Chassis, specify that the router replaces an existing trace file:

  [edit virtual-chassis]
  user@host# **set traceoptions file *filename* replace**

Related Documentation

- Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 337
- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Configuring Access to the Virtual Chassis Trace Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file, and you can explicitly set the default behavior of the log file.

To configure access to the trace log file for all users:

- On the router to be designated as the master of the Virtual Chassis, specify that all users can read the trace log file.

  [edit virtual-chassis]
  user@host# **set traceoptions file *filename* world-readable**

To explicitly set the default behavior to enable access to the trace log file only for the user who configured tracing:

- On the router to be designated as the master of the Virtual Chassis, specify that only the user who configured tracing can read the trace log file.

  [edit virtual-chassis]
  user@host# **set traceoptions file *filename* no-world-readable**

Related Documentation

- Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 337
- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Using Regular Expressions to Refine the Output of the Virtual Chassis Trace Log File

By default, the trace operation output includes all lines relevant to the logged events. You can refine the output of the trace log file for an MX Series Virtual Chassis by including regular expressions to be matched.

To refine the output of the trace log file:

- On the router to be designated as the master of the Virtual Chassis, configure a regular expression to be matched.

  [edit virtual-chassis]
  user@host# **set traceoptions file *filename* match *regular-expression***

Related Documentation

- Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 337
- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Configuring the Virtual Chassis Operations to Trace

By default, the router logs only important events. You can specify which operations to trace for an MX Series Virtual Chassis by including specific tracing flags when you configure tracing. Table 25 on page 342 describes the flags that you can include.

Table 25: Tracing Flags for MX Series Virtual Chassis

Flag	Description
all	Trace all operations.
auto-configuration	Trace Virtual Chassis ports that have been automatically configured.
csn	Trace Virtual Chassis complete sequence number (CSN) packets.
error	Trace Virtual Chassis errored packets.
graceful-restart	Trace Virtual Chassis graceful restart events.
hello	Trace Virtual Chassis hello packets.
krt	Trace Virtual Chassis kernel routing table (KRT) events.
lsp	Trace Virtual Chassis link-state packets.
lsp-generation	Trace Virtual Chassis link-state packet generation.
me	Trace Virtual Chassis mastership election (ME) events.
normal	Trace normal events.
packets	Trace Virtual Chassis packets.
parse	Trace reading of the configuration.
psn	Trace partial sequence number (PSN) packets.
route	Trace Virtual Chassis routing information.
spf	Trace Virtual Chassis shortest-path-first (SPF) events.
state	Trace Virtual Chassis state transitions.
task	Trace Virtual Chassis task operations.

To configure the flags for the Virtual Chassis operations to be logged:

1. Specify the tracing flag that represents the operation you want to trace.

   [edit virtual-chassis]
   user@host# **set traceoptions flag *flag***

2. (Optional) Specify one or more of the following additional tracing options for the specified flag:

   - To generate detailed trace output, use the **detail** option.
   - To disable a particular flag, use the **disable** option.
   - To trace received packets, use the **receive** option.
   - To trace transmitted packets, use the **send** option.

   For example, to generate detailed trace output for Virtual Chassis mastership election events in received packets:

   [edit virtual-chassis]
   user@host# **set traceoptions flag me detail receive**

Related Documentation

- Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 337
- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

CHAPTER 25

MX Series Virtual Chassis Configuration Examples

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

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

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

Requirements

This example uses the following software and hardware components:

- Junos OS Release 11.2

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

See Table 26 on page 347 for information about the hardware installed in each MX Series router.

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

Overview and Topology

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

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

This example configures a Virtual Chassis that interconnects two MX Series routers, and uses the basic topology shown in Figure 10 on page 347. For redundancy, two Virtual Chassis ports are configured on each member router.

Figure 10: Sample Topology for a Virtual Chassis with Two MX Series Routers

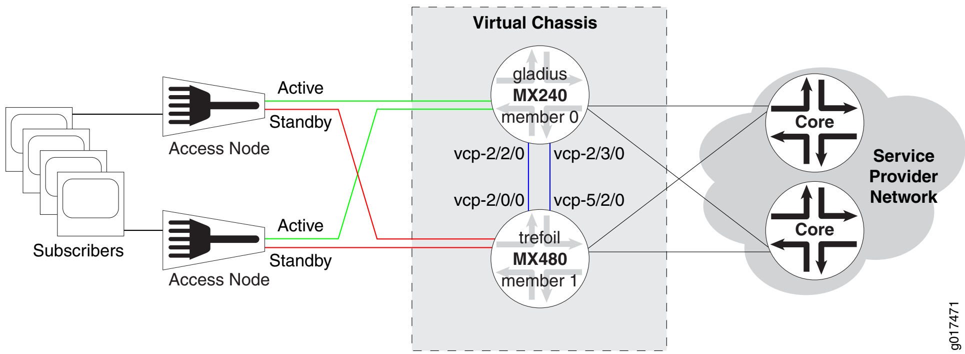


Table 26 on page 347 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis. You use some of these settings in the preprovisioned configuration and when you assign the member IDs and create the Virtual Chassis ports.

Table 26: 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 • Master RE-S-2000 Routing Engine in slot 0 (represented in example as **member0-re0**) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as **member0-re1**)	JN10C7135AFC	0	routing-engine (master)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 26: 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 • Master RE-S-2000 Routing Engine in slot 0 (represented in example as **member1-re0**) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as **member1-re1**)	JN115D117AFB	1	routing-engine (backup)	vcp-2/0/0 vcp-5/2/0	FPC 12 – 23 (offset = 12)

Configuration

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

- Preparing for the Virtual Chassis Configuration on page 349
- Creating and Applying Configuration Groups for the Virtual Chassis on page 351
- Configuring Preprovisioned Member Information for the Virtual Chassis on page 353
- Enabling Graceful Routing Engine Switchover and Nonstop Active Routing on page 354
- Configuring Member IDs and Rebooting the Routers to Enable Virtual Chassis Mode on page 355
- Configuring Virtual Chassis Ports to Interconnect Member Routers on page 357

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 master or backup of the Virtual Chassis.

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

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

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

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

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

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

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

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

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

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

Creating and Applying Configuration Groups for the Virtual Chassis

Step-by-Step Procedure

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

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

NOTE: The **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.

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

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

Results

Results Display the results of the configuration.

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



```
[edit]
user@gladius# show apply-groups
  apply-groups [ global member0-re0 member0-re1 member1-re0 member1-re1 ];
```

Configuring Preprovisioned Member Information for the Virtual Chassis

Step-by-Step Procedure

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

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

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

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

   ```
   [edit virtual-chassis]
   user@gladius# set member 0 role routing-engine serial-number JN10C7135AFC
   user@gladius# set member 1 role routing-engine serial-number JN115D117AFB
   ```

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

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

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

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

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

6. Commit the configuration.

Results

Results

Display the results of the configuration.

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

```
    }
    member 0 {
        role routing-engine;
        serial-number JN10C7135AFC;
    }
    member 1 {
        role routing-engine;
        serial-number JN115D117AFB;
    }
```

Enabling Graceful Routing Engine Switchover and Nonstop Active Routing

Step-by-Step Procedure

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

To enable graceful Routing Engine switchover and nonstop active routing:

1. Enable graceful Routing Engine switchover and nonstop active routing on member 0 (**gladius**):

   a. Log in to the console on member 0.

   b. Enable graceful switchover.

   ```
   [edit chassis redundancy]
   user@gladius# set graceful-switchover
   ```

   c. Enable nonstop active routing.

   ```
   [edit routing-options]
   user@gladius# set nonstop-routing
   ```

   d. Commit the configuration on member 0.

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

2. Enable graceful Routing Engine switchover and nonstop active routing on member 1 (**trefoil**):

   a. Log in to the console on member 1.

   b. Enable graceful switchover.

   ```
   [edit chassis redundancy]
   user@trefoil# set graceful-switchover
   ```

   c. Enable nonstop active routing.

   ```
   [edit routing-options]
   user@trefoil# set nonstop-routing
   ```

   d. Commit the configuration on member 1.

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


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

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

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

Step-by-Step Procedure

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

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

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

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

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

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

   Issuing the **request virtual-chassis member-id** command causes the router to reboot in preparation for membership in the Virtual Chassis.
3. Log in to the console on member 1 (**trefoil**).
4. Set the member ID on member 1.

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

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

Results

Results

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

For member 0 (**gladius**):

```
{master:member0-re0}

user@gladius> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: 4f2b.1aa0.de08
                                              Mastership           Neighbor List

Member ID       Status   Serial No    Model    priority    Role      ID  Interface

0 (FPC   0- 11) Prsnt    JN10C7135AFC mx240         129  Master*
```

For member 1 (**trefoil**):

```
Amnesiac (ttyd0)

login: user
Password:
...

{master:member1-re0}

user> show virtual-chassis status
Virtual Chassis ID: eabf.4e50.91e6
Virtual Chassis Mode: Disabled
                                              Mastership           Neighbor List

Member ID       Status   Serial No    Model    priority    Role      ID  Interface

1 (FPC  12- 23) Prsnt    JN115D117AFB mx480         128  Master*
```


Configuring Virtual Chassis Ports to Interconnect Member Routers

Step-by-Step Procedure

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

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

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

1. Configure the Virtual Chassis ports on member 0 (**gladius**).

   a. Log in to the console on member 0.

   b. Configure the first Virtual Chassis port that connects to member 1 (**trefoil**).

   ```
   {master:member0-re0}
   user@gladius> request virtual-chassis vc-port set fpc-slot 2 pic-slot 2 port 0
   vc-port successfully set
   ```

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

   c. When **vcp-2/2/0** is up, configure the second Virtual Chassis port that connects to member 1.

   ```
   {master:member0-re0}
   user@gladius> request virtual-chassis vc-port set fpc-slot 2 pic-slot 3 port 0
   vc-port successfully set
   ```

2. Configure the Virtual Chassis ports on member 1 (**trefoil**).

   a. Log in to the console on member 1.

   b. Configure the first Virtual Chassis port that connects to member 0 (**gladius**).

   ```
   {master:member1-re0}
   user@trefoil> request virtual-chassis vc-port set fpc-slot 2 pic-slot 0 port 0
   vc-port successfully set
   ```

   c. When **vcp-2/0/0** is up, configure the second Virtual Chassis port that connects to member 0.

   ```
   {master:member1-re0}
   user@trefoil> request virtual-chassis vc-port set fpc-slot 5 pic-slot 2 port 0
   ```

```
vc-port successfully set
```

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

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

```
{master:member0-re0}
user@gladius> show virtual-chassis status
{master:member0-re0}
user@gladius> show virtual-chassis vc-port all-members
```

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

4. Commit the configuration on the master router.

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

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

Verification

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

Verifying the Member IDs and Roles of the Virtual Chassis Members

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

Verifying the Operation of the Virtual Chassis Ports

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

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

```
{master:member0-re0}

user@gladius> show virtual-chassis vc-port all-members
member0:
--------------------------------------------------------------------------
Interface        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)
```

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

Related Documentation

- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Virtual Chassis Components Overview on page 288
- Guidelines for Configuring Virtual Chassis Ports on page 293
- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314

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

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

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

- Requirements on page 361
- Overview and Topology on page 361

Requirements

This example uses the following software and hardware components:

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

See Table 27 on page 362 for information about the hardware installed in each MX Series router.

Overview and Topology

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

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

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

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

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

This example deletes the Virtual Chassis configuration that uses the basic topology shown in Figure 11 on page 362. 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 11: Sample Topology for a Virtual Chassis with Two MX Series Routers

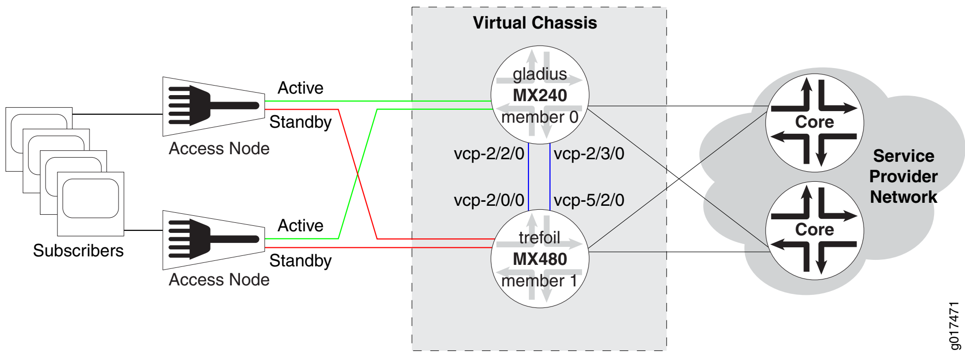


Table 27 on page 362 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 27: 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 • Master RE-S-2000 Routing Engine in slot 0 (represented in example as **member0-re0**) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as **member0-re1**)	JN10C7135AFC	0	routing-engine (master)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 27: 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 • Master RE-S-2000 Routing Engine in slot 0 (represented in example as **member1-re0**) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as **member1-re1**)	JN115D117AFB	1	routing-engine (backup)	vcp-2/0/0 vcp-5/2/0	FPC 12 – 23 (offset = 12)

Configuration

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

- Deleting Virtual Chassis Ports on page 364
- Deleting Configuration Group Definitions and Applications on page 365
- Deleting Preprovisioned Member Information on page 367
- Deleting Configured Interfaces on page 368
- Deleting Member IDs to Disable Virtual Chassis Mode on page 369

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

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

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]** hierararchy level on the master router and then propagated to the backup router during the formation of the Virtual Chassis.

To delete the preprovisioned member information for the Virtual Chassis:

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

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

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

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

Results

Results

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

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

Deleting Configured Interfaces

Step-by-Step Procedure

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

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

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

   a. Delete the configured interfaces.

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

   b. Commit the configuration on member 0.

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

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

   a. Delete the configured interfaces.

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

   b. Commit the configuration on member 1.

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

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

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

Deleting Member IDs to Disable Virtual Chassis Mode

Step-by-Step Procedure

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

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

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

1. In the console window on member 0 (**gladius**), delete the member ID and reboot the router.

   a. Exit configuration mode.

      {master:member0-re0}[edit]
      user@gladius# **exit**
      Exiting configuration mode

   b. Delete member ID **0**.

```
{master:member0-re0}

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

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

{master:member0-re0}
user@gladius>

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

System going down IMMEDIATELY
```

2. In the console window on member 1 (**trefoil**), delete the member ID and reboot the router.

   a. Exit configuration mode.

      {master:member1-re0}[edit]
      user@trefoil# **exit**
      Exiting configuration mode

   b. Delete member ID **1**.

```
{master:member1-re0}

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

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

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

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

System going down IMMEDIATELY
```

Results

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

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

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


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

Verification

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

- Verifying Deletion of the Virtual Chassis Ports on page 370
- Verifying Deletion of the Virtual Chassis Configuration Groups on page 371
- Verifying Deletion of the Virtual Chassis Member IDs on page 372

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

- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Virtual Chassis Components Overview on page 288
- Deleting a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 328

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

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

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

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

- Requirements on page 373
- Overview and Topology on page 373
- Configuration on page 374

Requirements

This example uses the following software and hardware and components:

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

See Table 28 on page 374 for information about the hardware installed in each MX Series router.

Overview and Topology

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

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

This example upgrades Junos OS in an MX Series Virtual Chassis configuration that uses the basic topology shown in Figure 12 on page 373. For redundancy, each member router is configured with two Virtual Chassis ports.

Figure 12: Sample Topology for a Virtual Chassis with Two MX Series Routers

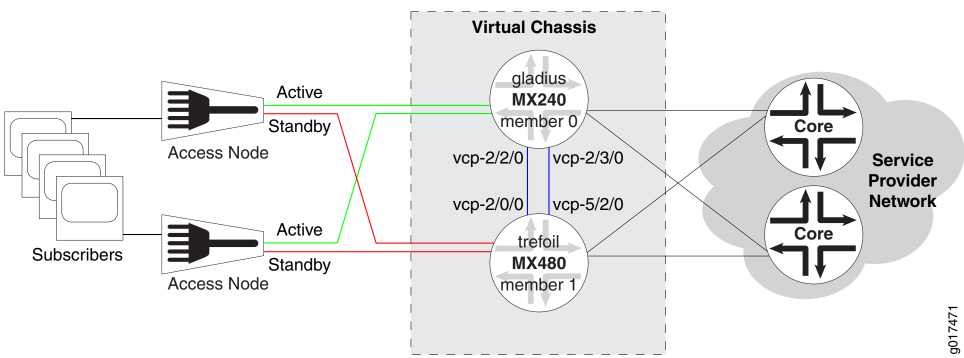


Table 28 on page 374 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 28: 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 • Master RE-S-2000 Routing Engine in slot 0 (represented in example as **member0-re0**) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as **member0-re1**)	JN10C7135AFC	0	routing-engine (master)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11
trefoil	MX480 router with: • Two 30-Gigabit Ethernet Queuing MPCs • Two 20-port Gigabit Ethernet MICs with SFP • Two 2-port 10-Gigabit Ethernet MICs with XFP • Master RE-S-2000 Routing Engine in slot 0 (represented in example as **member1-re0**) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as **member1-re1**)	JN115D117AFB	1	routing-engine (backup)	vcp-2/0/0 vcp-5/2/0	FPC 12 – 23 (offset = 12)

Configuration

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

- Preparing for the Upgrade on page 375
- Installing the Junos OS Software Package on Each Routing Engine on page 375

- Rebooting the Routing Engines to Run the New Junos OS Release on page 376
- Re-enabling Graceful Routing Engine Switchover and Nonstop Active Routing on page 377

Preparing for the Upgrade

Step-by-Step Procedure

To prepare for the upgrade process:

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

   See Downloading Software in the Installation and Upgrade Guide.

2. Disable nonstop active routing on the master router.

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

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

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

4. Commit the configuration on the master router.

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

5. Exit CLI configuration mode.

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

Installing the Junos OS Software Package on Each Routing Engine

Step-by-Step Procedure

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

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

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

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

   For example:

   ```
   {master:member0-re0}
   user@gladius> request system software add jinstall-11.2R1-8-domestic-signed.tgz
   ```

2. Install the software package on **member0-re1**.

   ```
   {master:member0-re0}
   user@gladius> request system software add package-name re1
   ```

3. Install the software package on **member1-re1**.

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

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

Results

Results

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

```
{master:member0-re0}

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


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

Rebooting the Routing Engines to Run the New Junos OS Release

Step-by-Step Procedure

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

1. Reboot **member1-re1**.

   ```
   {master:member0-re0}

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

   ```
   {master:member0-re0}

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

   ```
   {master:member0-re0}

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

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

Re-enabling Graceful Routing Engine Switchover and Nonstop Active Routing

Step-by-Step Procedure

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

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

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

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

   ```
   {master:member0-re0}[edit routing-options]
   user@gladius# set nonstop-routing
   ```
3. Commit the configuration on the master router.

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

Related Documentation

- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Virtual Chassis Components Overview on page 288
- Upgrading Junos OS in a Virtual Chassis Configuration for MX Series 3D Universal Edge Routers on page 331

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

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

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

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

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

Requirements

This example uses the following software and hardware and components:

- Junos OS Release 11.4
- One MX240 3D Universal Edge Router with dual Routing Engines
- One MX480 3D Universal Edge Router with dual Routing Engines

See Table 29 on page 379 for information about the hardware installed in each MX Series router.

Overview and Topology

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

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

Figure 13 on page 379 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 13: Sample Topology for a Virtual Chassis with Two MX Series Routers

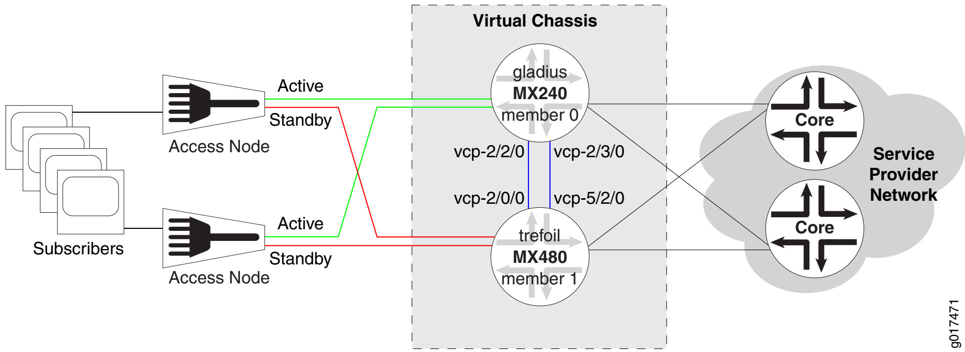


Table 29 on page 379 shows the hardware and software configuration settings for each MX Series router in the Virtual Chassis.

Table 29: 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 • Master RE-S-2000 Routing Engine in slot 0 (represented in example as **member0-re0**) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as **member0-re1**)	JN10C7135AFC	0	routing-engine (master)	vcp-2/2/0 vcp-2/3/0	FPC 0 – 11

Table 29: 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 • Master RE-S-2000 Routing Engine in slot 0 (represented in example as **member1-re0**) • Backup RE-S-2000 Routing Engine in slot 1 (represented in example as **member1-re1**)	JN115D117AFB	1	routing-engine (backup)	vcp-2/0/0 vcp-5/2/0	FPC 12 – 23 (offset = 12)

Configuration

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

- Removing the Routing Engine on page 381
- Returning the Routing Engine to Juniper Networks, Inc. on page 381
- Installing the New Routing Engine on page 381
- Installing the Junos OS Release on the New Routing Engine on page 381

Removing the Routing Engine

Step-by-Step Procedure

To remove the Routing Engine that needs repair or upgrade:

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

Returning the Routing Engine to Juniper Networks, Inc.

Step-by-Step Procedure

To return the Routing Engine to Juniper Networks, Inc:

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

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

Installing the New Routing Engine

Step-by-Step Procedure

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

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

Installing the Junos OS Release on the New Routing Engine

Step-by-Step Procedure

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

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

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

   See Downloading Software in the Installation and Upgrade Guide.

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

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

3. From the console, configure a plain-text password for the **root** (superuser) login.

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

4. From the console, commit the configuration.

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

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

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

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

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

   `{master:member0-re0}`

   `user@trefoil>` **request system software add member *member-id* re1 no-validate reboot *package-name* force**

   For example:

   `{master:member0-re0}`

   `user@trefoil>` **request system software add member 1 re1 no-validate reboot /var/tmp/jinstall-11.4R1-8-domestic-signed.tgz force**
   `Pushing bundle to re1...`

   This command reboots **member1-re1** after the software is added.

Results

Results After the reboot, the new Routing Engine becomes part of the Virtual Chassis, updates its command prompt to display **member1-re1**, and copies the appropriate configuration from the Virtual Chassis.

Verification

To verify that the MX Series Virtual Chassis is operating properly with the new Routing Engine, perform these tasks:

Verifying the Junos OS Installation on the New Routing Engine

Purpose Verify that you have installed the correct Junos OS release on the new Routing Engine (**member1-re1**).

Action Display the hostname, model name, and version information of the Junos OS release running on the new Routing Engine.

```
{local:member1-re1}

user@trefoil> show version local
Hostname: trefoil
Model: mx480
. . .
JUNOS Base OS boot [11.4R1-8]
JUNOS Base OS Software Suite [11.4R1-8]
. . .
```

Meaning The relevant portion of the **show version local** command output confirms that Junos OS Release 11.4R1-8 was installed as intended.

Verifying the Junos OS License Installation on the New Routing Engine

Purpose Verify that the MX Virtual Chassis Redundancy Feature Pack and the required Junos OS feature licenses are properly installed on the member router containing the new Routing Engine.

For information about license installation, see:

- Installing Junos OS Licenses on Virtual Chassis Member Routers on page 317
- Software Features That Require Licenses on MX Series Routers Only

Action Display the Junos OS licenses installed on the new Routing Engine.

```
{local:member1-re1}

user@trefoil> show system license
```

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

Meaning The **show system license** command output confirms that the MX Virtual Chassis Redundancy Feature Pack has been installed on this member router. In addition, the necessary Junos OS feature licenses have been installed to enable use of a particular software feature or scaling level.

Switching the Local Mastership in the Member Router to the New Routing Engine

Purpose Verify that the MX Series Virtual Chassis is operating properly with the new Routing Engine by confirming that the new Routing Engine can take over local mastership from the existing Routing Engine in the Virtual Chassis backup router, **trefoil** (member 1).

Action Switch the local mastership of the Routing Engines in **trefoil** from the Routing Engine in slot 0 (**member1-re0**) to the newly installed Routing Engine in slot 1 (**member1-re1**).

```
{backup:member1-re0}
user@trefoil> request chassis routing-engine master switch
```

Wait approximately 1 minute to display the status and roles of the member routers in the Virtual Chassis after the local switchover.

```
{backup:member1-re1}
user@trefoil> show virtual-chassis status
Preprovisioned Virtual Chassis
Virtual Chassis ID: a5b6.be0c.9525
                                              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 mastership of the Virtual Chassis. The output of the **show virtual-chassis status** command confirms that after the local switchover, member 0 (**gladius**) is still the Virtual Chassis master router, and member 1 (**trefoil**) is still the Virtual Chassis backup router.

Before the local switchover, **member1-re0** was the master Routing Engine in the Virtual Chassis backup router (VC-Bm), and **member1-re1** (the new Routing Engine) was the standby Routing Engine in the Virtual Chassis backup router (VC-Bs).

After the local switchover, **member1-re0** and **member1-re1** switch roles. The new Routing Engine, **member1-re1**, becomes the master Routing Engine in the Virtual Chassis backup router (VC-Bm), and **member1-re0** becomes the standby Routing Engine in the Virtual Chassis backup router (VC-Bs).

Table 30 on page 385 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 30 on page 385 apply only when you initiate the local switchover from the Virtual Chassis backup router (VC-B). For information about the role transitions that occur when you initiate the local switchover from the Virtual Chassis master router (VC-M), see "Switchover Behavior in a Virtual Chassis" on page 299.

Table 30: 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 master router (VC-M)	Virtual Chassis master router (VC-M)
trefoil (member 1)	Virtual Chassis backup router (VC-B)	Virtual Chassis backup router (VC-B)
member0-re0	Master Routing Engine in the Virtual Chassis master router (VC-Mm)	Master Routing Engine in the Virtual Chassis master router (VC-Mm)
member0-re1	Standby Routing Engine in the Virtual Chassis master router (VC-Ms)	Standby Routing Engine in the Virtual Chassis master router (VC-Ms)
member1-re0	Master Routing Engine in the Virtual Chassis backup router (VC-Bm)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)
member1-re1 (new Routing Engine)	Standby Routing Engine in the Virtual Chassis backup router (VC-Bs)	Master Routing Engine in the Virtual Chassis backup router (VC-Bm)

BEST PRACTICE: After you switch the local mastership of the Routing Engines, full synchronization of the Routing Engines takes approximately 30 minutes to complete. To prevent possible loss of subscriber state information due to incomplete synchronization, we recommend that you wait at least 30 minutes before performing another local switchover, global switchover, or graceful Routing Engine switchover in an MX Series Virtual Chassis configuration.

Related Documentation

- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Virtual Chassis Components Overview on page 288
- Installing Junos OS Licenses on Virtual Chassis Member Routers on page 317
- Switchover Behavior in a Virtual Chassis on page 299
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

- Installation and Upgrade Guide

Example: Configuring Class of Service for Virtual Chassis Ports on MX Series 3D Universal Edge Routers

This example illustrates a typical class of service (CoS) configuration that you might want to use for the Virtual Chassis ports in an MX Series Virtual Chassis.

Requirements

Before you begin:

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

  See “Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis” on page 345

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 31 on page 387 shows the nondefault CoS scheduler hierarchy configured in this example for the Virtual Chassis ports.

Table 31: Sample CoS Scheduler Hierarchy for Virtual Chassis Ports

Traffic Type	Queue Number	Priority	Transmit Rate/ Excess Rate
Network control (VCCP traffic)	3	Medium	90%
Expedited forwarding (voice traffic)	2	High	10%
Assured forwarding (video traffic)	1	Excess Low	99%
Best effort (data traffic)	0	Excess Low	1%

In this example, you create a nondefault CoS configuration for Virtual Chassis ports by completing the following tasks on the Virtual Chassis master router:

- Associate forwarding classes with **queue 0** through **queue 3**, and configure a fabric priority value for each queue.
- Configure an output traffic control profile named **tcp-vcp-ifd** to define traffic scheduling parameters, and associate a scheduler map named **sm-vcp-ifd** with the traffic control profile.
- Apply the output traffic-control profile to the **vcp-*** interface, which represents all Virtual Chassis port interfaces in the Virtual Chassis.
- Associate the **sm-vcp-ifd** scheduler map with the forwarding classes and scheduler configuration.
- Configure the parameters for schedulers **s-medium-priority**, **s-high-priority**, **s-low-priority**, **s-high-weight**, and **s-low-weight**.

Configuration

CLI Quick Configuration

To quickly create a nondefault CoS configuration for Virtual Chassis ports, copy the following commands and paste them into the router terminal window:

```
[edit]
set class-of-service forwarding-classes queue 0 best-effort
set class-of-service forwarding-classes queue 0 priority low
set class-of-service forwarding-classes queue 1 assured-forwarding
set class-of-service forwarding-classes queue 1 priority low
```

```
set class-of-service forwarding-classes queue 2 expedited-forwarding
set class-of-service forwarding-classes queue 2 priority high
set class-of-service forwarding-classes queue 3 network-control
set class-of-service forwarding-classes queue 3 priority high
set class-of-service traffic-control-profiles tcp-vcp-ifd scheduler-map sm-vcp-ifd
set class-of-service interfaces vcp-* output-traffic-control-profile tcp-vcp-ifd
set class-of-service scheduler-maps sm-vcp-ifd forwarding-class network-control
  scheduler s-medium-priority
set class-of-service scheduler-maps sm-vcp-ifd forwarding-class expedited-forwarding
  scheduler s-high-priority
set class-of-service scheduler-maps sm-vcp-ifd forwarding-class assured-forwarding
  scheduler s-high-weight
set class-of-service scheduler-maps sm-vcp-ifd forwarding-class best-effort scheduler
  s-low-weight
set class-of-service schedulers s-medium-priority transmit-rate percent 10
set class-of-service schedulers s-medium-priority priority medium-high
set class-of-service schedulers s-medium-priority excess-priority high
set class-of-service schedulers s-high-priority transmit-rate percent 90
set class-of-service schedulers s-high-priority priority high
set class-of-service schedulers s-high-priority excess-priority high
set class-of-service schedulers s-low-priority priority low
set class-of-service schedulers s-high-weight excess-rate percent 99
set class-of-service schedulers s-low-weight excess-rate percent 1
```

Step-by-Step Procedure

To create a nondefault CoS configuration for Virtual Chassis ports in an MX Series Virtual Chassis:

1. Log in to the console on the master router of the Virtual Chassis.
2. Specify that you want to configure CoS forwarding classes.

   ```
   {master:member0-re0} [edit]
   user@host# edit class-of-service forwarding-classes
   ```

3. Associate a forwarding class with each queue name and number, and configure a fabric priority value for each queue.

   ```
   {master:member0-re0} [edit class-of-service forwarding-classes]
   user@host# set queue 0 best-effort priority low
   user@host# set queue 1 assured-forwarding priority low
   user@host# set queue 2 expedited-forwarding priority high
   user@host# set queue 3 network-control priority high
   ```

4. Return to the **[edit class-of-service]** hierarchy level to configure an output traffic-control profile.

   ```
   {master:member0-re0} [edit class-of-service forwarding-classes]
   user@host# up
   ```

5. Configure an output traffic-control profile and associate it with a scheduler map.

   ```
   {master:member0-re0} [edit class-of-service]
   user@host# set traffic-control-profiles tcp-vcp-ifd scheduler-map sm-vcp-ifd
   ```

6. Apply the output traffic-control profile to all Virtual Chassis port interfaces in the Virtual Chassis.

   ```
   {master:member0-re0} [edit class-of-service]
   user@host# set interfaces vcp-* output-traffic-control-profile tcp-vcp-ifd
   ```

7. Specify that you want to configure the scheduler map.

```
{master:member0-re0} [edit class-of-service]
user@host# edit scheduler-maps sm-vcp-ifd
```

8. Associate the scheduler map with the scheduler configuration and forwarding classes.

```
{master:member0-re0} [edit class-of-service scheduler-maps sm-vcp-ifd]
user@host# set forwarding-class network-control scheduler s-medium-priority
user@host# set forwarding-class expedited-forwarding scheduler s-high-priority
user@host# set forwarding-class assured-forwarding scheduler s-high-weight
user@host# set forwarding-class best-effort scheduler s-low-weight
```

9. Return to the **[edit class-of-service]** hierarchy level to configure the schedulers.

```
{master:member0-re0} [edit class-of-service scheduler-maps sm-vcp-ifd]
user@host# up 2
```

10. Configure parameters for the schedulers.

```
{master:member0-re0} [edit class-of-service]
user@host# set schedulers s-medium-priority priority medium-high excess-priority
  high transmit-rate percent 10
user@host# set schedulers s-high-priority priority high excess-priority high
  transmit-rate percent 90
user@host# set schedulers s-low-priority priority low
user@host# set schedulers s-high-weight excess-rate percent 99
user@host# set schedulers s-low-weight excess-rate percent 1
```

Results

Results From the **[edit class-of-service]** hierarchy level in configuration mode, confirm the results of your configuration by issuing the **show** statement. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
{master:member0-re0} [edit class-of-service]
user@host# show
forwarding-classes {
  queue 0 best-effort priority low;
  queue 1 assured-forwarding priority low;
  queue 2 expedited-forwarding priority high;
  queue 3 network-control priority high;
}
traffic-control-profiles {
  tcp-vcp-ifd {
    scheduler-map sm-vcp-ifd;
  }
}
interfaces {
  vcp-* {
    output-traffic-control-profile tcp-vcp-ifd;
  }
}
scheduler-maps {
  sm-vcp-ifd {
```

```
            forwarding-class network-control scheduler s-medium-priority;
            forwarding-class expedited-forwarding scheduler s-high-priority;
            forwarding-class assured-forwarding scheduler s-high-weight;
            forwarding-class best-effort scheduler s-low-weight;
        }
    }
    schedulers {
        s-medium-priority {
            transmit-rate percent 10;
            priority medium-high;
            excess-priority high;
        }
        s-high-priority {
            transmit-rate percent 90;
            priority high;
            excess-priority high;
        }
        s-low-priority {
            priority low;
        }
        s-high-weight {
            excess-rate percent 99;
        }
        s-low-weight {
            excess-rate percent 1;
        }
    }
```

If you are done configuring CoS on the master router, enter **commit** from configuration mode.

Related Documentation

- Class of Service Overview for Virtual Chassis Ports on page 305
- Guidelines for Configuring Class of Service for Virtual Chassis Ports on page 310
- Junos OS Class of Service Configuration Guide

CHAPTER 26

Verifying and Managing MX Series Virtual Chassis Configurations

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 32 on page 392 describes the commands that you can run on all (both) member routers (with the **all-members** option), on the local member router (with the **local** option), or on a specific member router (with the **member *member-id*** option) in an MX Series Virtual Chassis configuration.

Table 32: Commands Available for Command Forwarding in an MX Series Virtual Chassis

Command	Purpose	all-members	local	member *member-id*
request chassis fpc	Control the operation of the Flexible PIC Concentrator (FPC).	Change FPC status of all members of the Virtual Chassis configuration.	(Default) Change FPC status of the local Virtual Chassis member.	Change FPC status of the specified member of the Virtual Chassis configuration.
request chassis fpm resync	Resynchronize the craft interface status.	Resynchronize the craft interface status on all members of the Virtual Chassis configuration.	(Default) Resynchronize the craft interface status on the local Virtual Chassis member.	Resynchronize the craft interface status on the specified member of the Virtual Chassis configuration.
request chassis routing-engine master	Control which Routing Engine is the master for a router with dual Routing Engines.	Control Routing Engine mastership on the Routing Engines in all member routers of the Virtual Chassis configuration.	(Default) Control Routing Engine mastership on the Routing Engines in the local Virtual Chassis configuration.	Control Routing Engine mastership on the Routing Engines of the specified member in the Virtual Chassis configuration.
request routing-engine login	Specify a **tty** connection for login for a router with two Routing Engines.	Log in to all members of the Virtual Chassis configuration.	(Default) Log in to the local Virtual Chassis member.	Log in to the specified member of the Virtual Chassis configuration.
request support information	Display information about the system.	(Default) Display system information for all members of the Virtual Chassis configuration.	Display system information for the local Virtual Chassis member.	Display system information for the specified member of the Virtual Chassis configuration.
request system halt	Stop the router.	(Default) Halt all members of the Virtual Chassis configuration.	Halt the local Virtual Chassis member.	Halt the specified member of the Virtual Chassis configuration.
request system partition abort	Terminate a previously scheduled storage media partition operation.	(Default) Abort a previously scheduled storage media partition operation for all members of the Virtual Chassis configuration.	Abort a previously scheduled storage media partition operation for the local Virtual Chassis member.	Abort a previously scheduled storage media partition operation for the specified member of the Virtual Chassis member.
request system partition hard-disk	Set up the hard disk for partitioning.	(Default) Schedule a partition of the hard disk for all members of the Virtual Chassis configuration.	Schedule a partition of the hard disk for the local Virtual Chassis member.	Schedule a partition of the hard disk for the specified member of the Virtual Chassis configuration.

Table 32: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(continued)*

Command	Purpose	all-members	local	member *member-id*
request system power-off	Power off the software.	(Default) Power off all members of the Virtual Chassis configuration.	Power off the local Virtual Chassis member.	Power off the specified member of the Virtual Chassis configuration.
request system reboot	Reboot the software.	(Default) Reboot the software on all members of the Virtual Chassis configuration.	Reboot the software on the local Virtual Chassis member.	Reboot the software on the specified member of the Virtual Chassis configuration.
request system snapshot	Back up the currently running and active file system partitions on the router to standby partitions that are not running.	(Default) Archive data and executable areas for all members of the Virtual Chassis configuration.	Archive data and executable areas for the local Virtual Chassis member.	Archive data and executable areas for the specified member of the Virtual Chassis configuration.
request system software add	Install a software package or bundle on the router.	(Default if no options specified) Install a software package on all members of the Virtual Chassis configuration.	—	Install a software package on the specified member of the Virtual Chassis configuration.
request system software rollback	Revert to the software that was loaded at the last successful **request system software add** command.	(Default) Attempt to roll back to the previous set of packages on all members of the Virtual Chassis configuration.	Attempt to roll back to the previous set of packages on the local Virtual Chassis member.	Attempt to roll back to the previous set of packages on the specified member of the Virtual Chassis configuration.
request system software validate	Validate candidate software against the current configuration of the router.	—	(Default if no options specified) Validate the software package on the local Virtual Chassis member.	Validate the software bundle or package on the specified member of the Virtual Chassis configuration.
request system storage cleanup	Free storage space on the router or switch by rotating log files and proposing a list of files for deletion.	(Default) Delete files on all members of the Virtual Chassis configuration.	Delete files on the local Virtual Chassis member.	Delete files on the specified member of the Virtual Chassis configuration.
restart	Restart a Junos OS process.	Restart the software process for all members of the Virtual Chassis configuration.	(Default) Restart the software process for the local Virtual Chassis member.	Restart the software process for a specified member of the Virtual Chassis configuration.
show chassis alarms	Display information about the conditions that have been configured to trigger alarms.	(Default) Display information about alarm conditions for all the member routers of the Virtual Chassis configuration.	Display information about alarm conditions for the local Virtual Chassis member.	Display information about alarm conditions for the specified member of the Virtual Chassis configuration.

Table 32: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(continued)*

Command	Purpose	all-members	local	member *member-id*
show chassis craft-interface	View messages currently displayed on the craft interface.	(Default) Display information currently on the craft interface for all members of the Virtual Chassis configuration.	Display information currently on the craft interface for the specified member of the Virtual Chassis configuration.	Display information currently on the craft interface for the specified member of the Virtual Chassis configuration.
show chassis environment	Display environmental information about the router or switch chassis, including the temperature and information about the fans, power supplies, and Routing Engine.	(Default) Display chassis environmental information for all the members of the Virtual Chassis configuration.	Display chassis environmental information for the local Virtual Chassis member.	Display chassis environmental information for the specified member of the Virtual Chassis configuration.
show chassis environment cb	Display environmental information about the Control Boards (CBs).	(Default) Display environmental information about the CBs on all the members of the Virtual Chassis configuration.	Display environmental information about the CBs on the local Virtual Chassis member.	Display environmental information about the CBs on the specified member of the Virtual Chassis configuration.
show chassis environment fpc	Display environmental information about Flexible PIC Concentrators (FPCs).	(Default) Display environmental information for the FPCs in all the members of the Virtual Chassis configuration.	Display environmental information for the FPCs in the local Virtual Chassis member.	Display environmental information for the FPCs in the specified member of the Virtual Chassis configuration.
show chassis environment pem	Display Power Entry Module (PEM) environmental status information.	(Default) Display environmental information about the PEMs in all the member routers of the Virtual Chassis configuration.	Display environmental information about the PEMs in the local Virtual Chassis member.	Display environmental information about the PEMs in the specified member of the Virtual Chassis configuration.
show chassis environment routing-engine	Display Routing Engine environmental status information.	(Default) Display environmental information about the Routing Engines in all member routers in the Virtual Chassis configuration.	Display environmental information about the Routing Engines in the local Virtual Chassis member.	Display environmental information about the Routing Engines in the specified member of the Virtual Chassis configuration.
show chassis ethernet-switch	Display information about the ports on the Control Board (CB) Ethernet switch.	(Default) Display information about the ports on the CB Ethernet switch on all the members of the Virtual Chassis configuration.	Display information about the ports on the CB Ethernet switch on the local Virtual Chassis member.	Display information about the ports on the CB Ethernet switch on the specified member of the Virtual Chassis configuration.

Table 32: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(continued)*

Command	Purpose	all-members	local	member *member-id*
show chassis fabric fpcs	Display the state of the electrical and optical switch fabric links between the Flexible PIC Concentrators (FPCs) and the Switch Interface Boards (SIBs).	(Default) Display the switching fabric link states for the FPCs in all members of the Virtual Chassis configuration.	Display the switching fabric link states for the FPCs in the local Virtual Chassis member.	Display the switching fabric link states for the FPCs in the specified member of the Virtual Chassis configuration.
show chassis fabric map	Display the switching fabric map state.	(Default) Display the switching fabric map state for all the members of the Virtual Chassis configuration.	Display the switching fabric map state for the local Virtual Chassis member.	Display the switching fabric map state for the specified member of the Virtual Chassis configuration.
show chassis fabric plane	Display the state of all fabric plane connections.	(Default) Display the state of all fabric plane connections on all members of the Virtual Chassis configuration.	Display the state of all fabric plane connections on the local Virtual Chassis member.	Display the state of all fabric plane connections on the specified member of the Virtual Chassis configuration.
show chassis fabric plane-location	Display the Control Board (CB) location of each plane on both the master and backup Routing Engine.	(Default) Display the CB location of each fabric plane on the Routing Engines in all member routers in the Virtual Chassis configuration.	Display the CB location of each fabric plane on the Routing Engines in the local Virtual Chassis member.	Display the CB location of each fabric plane on the Routing Engines in the specified member in the Virtual Chassis configuration.
show chassis fan	Display information about the fan tray and fans.	(Default) Display information about the fan tray and fans for all members of the Virtual Chassis configuration.	Display information about the fan tray and fans for the local Virtual Chassis member.	Display information about the fan tray and fans for the specified member of the Virtual Chassis configuration.
show chassis firmware	Display the version levels of the firmware running on the System Control Board (SCB), Switching and Forwarding Module (SFM), System and Switch Board (SSB), Forwarding Engine Board (FEB), Flexible PIC Concentrators (FPCs), and Routing Engines.	(Default) Display the version levels of the firmware running for all members of the Virtual Chassis configuration.	Display the version levels of the firmware running for the local Virtual Chassis member.	Display the version levels of the firmware running for the specified member of the Virtual Chassis configuration.

Table 32: 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 Virtual Chassis member.	Display status information for all FPCs on the specified member of the Virtual Chassis configuration.
show chassis hardware	Display a list of all Flexible PIC Concentrators (FPCs) and PICs installed in the router or switch chassis, including the hardware version level and serial number.	(Default) Display hardware-specific information for all the members of the Virtual Chassis configuration.	Display hardware-specific information for the local Virtual Chassis member.	Display hardware-specific information for the specified member of the Virtual Chassis configuration.
show chassis location	Display the physical location of the chassis.	(Default) Display the physical location of the chassis for all the member routers in the Virtual Chassis configuration.	Display the physical location of the chassis for the local Virtual Chassis member.	Display the physical location of the chassis for the specified member of the Virtual Chassis configuration.
show chassis mac-addresses	Display the media access control (MAC) addresses for the router, switch chassis, or switch.	(Default) Display the MAC addresses for all the member routers of the Virtual Chassis configuration.	Display the MAC addresses for the local Virtual Chassis member.	Display the MAC addresses for the specified member of the Virtual Chassis configuration.
show chassis pic	Display status information about the PIC installed in the specified Flexible PIC Concentrator (FPC) and PIC slot.	(Default) Display PIC information for all member routers in the Virtual Chassis configuration.	Display PIC information for the local Virtual Chassis member.	Display PIC information for the specified member of the Virtual Chassis configuration.
show chassis power	Display power limits and usage information for the AC or DC power sources.	(Default) Display power usage information for all members of the Virtual Chassis configuration.	Display power usage information for the local Virtual Chassis member.	Display power usage information for the specified member of the Virtual Chassis configuration.
show chassis routing-engine	Display the status of the Routing Engine.	(Default) Display Routing Engine information for all members of the Virtual Chassis configuration.	Display Routing Engine information for the local Virtual Chassis member.	Display Routing Engine information for the specified member of the Virtual Chassis configuration.
show chassis temperature-thresholds	Display chassis temperature threshold settings, in degrees Celsius.	(Default) Display the chassis temperature threshold settings of all member routers in the Virtual Chassis configuration.	Display the chassis temperature threshold settings of the local Virtual Chassis member.	Display the chassis temperature threshold settings of the specified member of the Virtual Chassis configuration.

Table 32: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(continued)*

Command	Purpose	all-members	local	member *member-id*
show pfe fpc	Display Packet Forwarding Engine statistics for the specified Flexible PIC Concentrator (FPC).	(Default) Display Packet Forwarding Engine statistics for the specified FPC in all members of the Virtual Chassis configuration.	Display Packet Forwarding Engine statistics for the specified FPC in the local Virtual Chassis member.	Display Packet Forwarding Engine statistics for the specified FPC in the specified member of the Virtual Chassis configuration.
show pfe terse	Display Packet Forwarding Engine status information.	(Default) Display Packet Forwarding Engine status information for all members in the Virtual Chassis configuration.	Display Packet Forwarding Engine status information for the local Virtual Chassis member.	Display Packet Forwarding Engine status information for the specified member of the Virtual Chassis configuration.
show system audit	Display the state and checksum values for file systems.	(Default) Display file system MD5 hash and permissions information on all members of the Virtual Chassis configuration.	Display file system MD5 hash and permissions information on the local Virtual Chassis member.	Display file system MD5 hash and permissions information on the specified member of the Virtual Chassis configuration.
show system boot-messages	Display initial messages generated by the system kernel upon startup.	(Default) Display boot time messages on all members of the Virtual Chassis configuration.	Display boot time messages on the local Virtual Chassis member.	Display boot time messages on the specified member of the Virtual Chassis configuration.
show system buffers	Display information about the buffer pool that the Routing Engine uses for local traffic.	(Default) Show buffer statistics for all members of the Virtual Chassis configuration.	Show buffer statistics for the local Virtual Chassis member.	Show buffer statistics for the specified member of the Virtual Chassis configuration.
show system connections	Display information about the active IP sockets on the Routing Engine.	(Default) Display system connection activity for all members of the Virtual Chassis configuration.	Display system connection activity for the local Virtual Chassis member.	Display system connection activity for the specified member of the Virtual Chassis configuration.
show system directory-usage	Display directory usage information.	Display directory information for all members of the Virtual Chassis configuration.	(Default) Display directory information for the local Virtual Chassis member.	Display directory information for the specified member of the Virtual Chassis configuration.
show system processes	Display information about software processes that are running on the router and that have controlling terminals.	(Default) Display standard system process information for all members of the Virtual Chassis configuration.	Display standard system process information for the local Virtual Chassis member.	Display standard system process information for the specified member of the Virtual Chassis configuration.

Table 32: Commands Available for Command Forwarding in an MX Series Virtual Chassis *(continued)*

Command	Purpose	all-members	local	member *member-id*
show system queues	Display queue statistics.	(Default) Display system queue statistics for all members of the Virtual Chassis configuration.	Display system queue statistics for the local Virtual Chassis member.	Display system queue statistics for the specified member of the Virtual Chassis configuration.
show system reboot	Display pending system reboots or halts.	(Default) Display halt or reboot request information for all members of the Virtual Chassis configuration.	Display halt or reboot request information for the local Virtual Chassis member.	Display halt or reboot request information for the specified member of the Virtual Chassis configuration.
show system statistics	Display system-wide protocol-related statistics.	(Default) Display system statistics for a protocol for all members of the Virtual Chassis configuration.	Display system statistics for a protocol for the local Virtual Chassis member.	Display system statistics for a protocol for the specified member of the Virtual Chassis configuration.
show system storage	Display statistics about the amount of free disk space in the router's file systems.	(Default) Display system storage statistics for all members of the Virtual Chassis configuration.	Display system storage statistics for the local Virtual Chassis member.	Display system storage statistics for the specified member of the Virtual Chassis configuration.
show system switchover	Display whether graceful Routing Engine switchover is configured, the state of the kernel replication (ready or synchronizing), any replication errors, and whether the primary and standby Routing Engines are using compatible versions of the kernel database.	(Default) Display graceful Routing Engine switchover information for all Routing Engines on all members of the Virtual Chassis configuration.	Display graceful Routing Engines switchover information for all Routing Engines on the local Virtual Chassis member.	Display graceful Routing Engine switchover information for all Routing Engines on the specified member of the Virtual Chassis configuration.
show system uptime	Display the current time and information about how long the router or switch, router or switch software, and routing protocols have been running.	(Default) Show time since the system rebooted and processes started on all members of the Virtual Chassis configuration.	Show time since the system rebooted and processes started on the local Virtual Chassis member.	Show time since the system rebooted and processes started on the specified member of the Virtual Chassis configuration.
show system users	List information about the users who are currently logged in to the router.	(Default) Display users currently logged in to all members of the Virtual Chassis configuration.	Display users currently logged in to the local Virtual Chassis member.	Display users currently logged in to the specified member of the Virtual Chassis configuration.

Table 32: 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 Virtual Chassis member.	Display kernel dynamic memory usage information for the specified member of the Virtual Chassis configuration.
show version	Display the hostname and version information about the software running on the router.	(Default) Display standard information about the hostname and version of the software running on all members of the Virtual Chassis configuration.	Display standard information about the hostname and version of the software running on the local Virtual Chassis member.	Display standard information about the hostname and version of the software running on the specified member of the Virtual Chassis configuration.
show version invoke-on	Display the hostname and version information about the software running on a router with two Routing Engines.	(Default) Display the hostname and version information about the software running on all master and backup Routing Engines on all members of the Virtual Chassis configuration.	Display the hostname and version information about the software running on all master and backup Routing Engines on the local Virtual Chassis member.	Display the hostname and version information about the software running on all master and backup Routing Engines on the specified member of the Virtual Chassis configuration.

Related Documentation

- Virtual Chassis Components Overview on page 288
- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314

- Junos OS Operational Mode Commands

Virtual Chassis Slot Number Mapping for Use with SNMP

Junos OS supports the use of SNMP to monitor the routers and other devices in your network. For example, the Juniper Networks jnxBoxAnatomy enterprise-specific Chassis MIB contains the jnxFruTable object, which shows the status of field-replaceable units (FRUs) in the chassis. Within the jnxFruTable object, the jnxFruSlot object displays the slot number where the FRU is installed.

If you are using the jnxFruSlot object in jnxFruTable to display the slot numbers of line cards installed in a member router of an MX Series Virtual Chassis, keep in mind that the offset used for slot numbering in an MX Series Virtual Chassis affects the value that appears for the jnxFruSlot object.

Table 33 on page 400 lists the jnxFruSlot number that appears in the jnxFruTable of the jnxBoxAnatomy MIB, and the corresponding line card physical slot number in each member router of a two-member MX Series Virtual Chassis. For example, a jnxFruSlot value of 15 corresponds to physical slot 3 in member 0 of an MX Series Virtual Chassis. A jnxFruSlot value of 30 corresponds to physical slot 6 in member 1 of an MX Series Virtual Chassis.

Table 33: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series Virtual Chassis

jnxFruSlot Number	Line Card Slot Number	MX Series Virtual Chassis Member ID
Line Cards in MX Series Virtual Chassis Member ID 0 (offset = 12):		
12	0	0
13	1	0
14	2	0
15	3	0
16	4	0
17	5	0
18	6	0
19	7	0
20	8	0
21	9	0
22	10	0
23	11	0
Line Cards in MX Series Virtual Chassis Member ID 1 (offset = 24)		
24	0	1
25	1	1
26	2	1
27	3	1
28	4	1
29	5	1
30	6	1
31	7	1
32	8	1

Table 33: jnxFruSlot Numbers and Corresponding Slot Numbers in an MX Series Virtual Chassis *(continued)*

jnxFruSlot Number	Line Card Slot Number	MX Series Virtual Chassis Member ID
33	9	1
34	10	1
35	11	1

Related Documentation

- Virtual Chassis Components Overview on page 288
- SNMP MIBs and Traps Reference

Managing Files on Virtual Chassis Member Routers

In a Virtual Chassis configuration for MX Series 3D Universal Edge Routers, you can manage files on local and remote member routers by including a member specification in the following **file** operational commands:

file archive	**file copy**
file checksum md5	**file delete**
file checksum sha1	**file list**
file checksum sha-256	**file rename**
file compare	**file show**

The member specification identifies the specific Virtual Chassis member router and Routing Engine on which you want to manage files, and includes both of the following elements:

- The Virtual Chassis member ID (**0** or **1**)
- The Routing Engine slot number (**re0** or **re1**)

To manage files on a specific member router and a specific Routing Engine in an MX Series Virtual Chassis:

- From operational mode, issue the **file** command and Virtual Chassis member specification:

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

  `user@host>` **file** ***option*** **member(0 | 1)-re(0 | 1):*command-option***

  For example, the following **file list** command uses the **member1-re0** specification to display a list of the files in the **/config** directory on the Routing Engine in slot 0 (**re0**) in Virtual Chassis **member 1**. The router forwards the command from **member 0**, where it is issued, to **member 1**, and displays the results as if the command were processed on the local router.

```
{master:member0-re0}

user@host> file list member1-re0:/config
member1-re0:
--------------------------------------------------------------------------

/config:
.snap/
juniper.conf.1.gz
juniper.conf.2.gz
juniper.conf.3.gz
juniper.conf.gz
juniper.conf.md5
license/
license.old/
usage.db
vchassis/
```

Related Documentation

- Interchassis Redundancy and Virtual Chassis Overview on page 285
- Virtual Chassis Components Overview on page 288
- Format for Specifying Filenames and URLs in Junos OS CLI Commands in the Junos OS System Basics Configuration Guide

Verifying the Status of Virtual Chassis Member Routers

Purpose Verify that the member routers in an MX Series Virtual Chassis are properly configured.

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

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

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Verifying the Operation of Virtual Chassis Ports

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

Action

- To display the status of the Virtual Chassis ports for both member routers in the Virtual Chassis:

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

- To display the status of the Virtual Chassis ports for a specified member router in the Virtual Chassis:

  ```
  user@host> show virtual-chassis vc-port member member-id
  ```


- To display the status of the Virtual Chassis ports for the member router on which you are issuing the command:

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

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Verifying Neighbor Reachability for Member Routers in a Virtual Chassis

Purpose Verify that each member router in an MX Series Virtual Chassis has a path to reach the neighbor routers to which it is connected.

Action

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

  `user@host>` **show virtual-chassis active-topology all-members**
- To display neighbor reachability information for a specified member router in the Virtual Chassis:

  `user@host>` **show virtual-chassis active-topology member *member-id***
- To display neighbor reachability information for the member router on which you are issuing the command:

  `user@host>` **show virtual-chassis active-topology local**

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Verifying Neighbor Reachability for Hardware Devices in a Virtual Chassis

Purpose Verify that each hardware device in a member router in an MX Series Virtual Chassis can reach the neighbor routers and devices to which it is connected. On the MX Series routing platform, there is only one active device for each member router.

Action

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

  `user@host>` **show virtual-chassis device-topology all-members**
- To display neighbor reachability information for the device in a specified member router in the Virtual Chassis:

  `user@host>` **show virtual-chassis device-topology member *member-id***
- To display neighbor reachability information for the device in the member router on which you are issuing the command:

  `user@host>` **show virtual-chassis device-topology local**

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Viewing Information in the Virtual Chassis Control Protocol Adjacency Database

Purpose View information about neighbors in the Virtual Chassis Control Protocol (VCCP) adjacency database for an MX Series Virtual Chassis configuration.

Action

- To display VCCP neighbor adjacency information for both member routers in the Virtual Chassis:

  `user@host>` **show virtual-chassis protocol adjacency all-members**
- To display VCCP neighbor adjacency information for a specified member router in the Virtual Chassis:

  `user@host>` **show virtual-chassis protocol adjacency member *member-id***
- To display VCCP neighbor adjacency information for the device with a specified system ID:

  `user@host>` **show virtual-chassis protocol adjacency *system-id***
- To display VCCP neighbor adjacency information for the device with a specified system ID on the specified member router:

  `user@host>` **show virtual-chassis protocol adjacency member *member-id system-id***
- To display VCCP neighbor adjacency information for the member router on which you are issuing the command:

  `user@host>` **show virtual-chassis protocol adjacency local**

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Viewing Information in the Virtual Chassis Control Protocol Link-State Database

Purpose View information about protocol data unit (PDU) packets in the Virtual Chassis Control Protocol (VCCP) link-state database for an MX Series Virtual Chassis configuration.

Action

- To display VCCP PDU information for both member routers in the Virtual Chassis:

  `user@host>` **show virtual-chassis protocol database all-members**
- To display VCCP PDU information for a specified member router in the Virtual Chassis:

  `user@host>` **show virtual-chassis protocol database member *member-id***
- To display VCCP PDU information for the device with a specified system ID:

  `user@host>` **show virtual-chassis protocol database *system-id***

- To display VCCP PDU information for the device with a specified system ID on the specified member router:

  `user@host>` **show virtual-chassis protocol database member** ***member-id system-id***

- To display VCCP PDU information for the member router on which you are issuing the command:

  `user@host>` **show virtual-chassis protocol database local**

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Viewing Information About Virtual Chassis Port Interfaces in the Virtual Chassis Control Protocol Database

Purpose View information in the Virtual Chassis Control Protocol (VCCP) database about Virtual Chassis port interfaces in an MX Series Virtual Chassis.

Action

- To display VCCP information about Virtual Chassis port interfaces for both member routers:

  `user@host>` **show virtual-chassis protocol interface all-members**

- To display VCCP information about Virtual Chassis port interfaces for a specified member router:

  `user@host>` **show virtual-chassis protocol interface member** ***member-id***

- To display VCCP information about a specified Virtual Chassis port interface:

  `user@host>` **show virtual-chassis protocol interface vcp-*slot/pic/port.logical-unit-number***

- To display VCCP information about Virtual Chassis port interfaces for the member router on which you are issuing the command:

  `user@host>` **show virtual-chassis protocol interface local**

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Viewing Virtual Chassis Control Protocol Routing Tables

Purpose View Virtual Chassis Control Protocol (VCCP) unicast and multicast routing tables for an MX Series Virtual Chassis configuration.

Action

- 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 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

Viewing Virtual Chassis Control Protocol Statistics for Member Routers and Virtual Chassis Ports

Purpose View Virtual Chassis Control Protocol (VCCP) statistics for one or both member routers, or for a specified Virtual Chassis port interface, in an MX Series Virtual Chassis configuration.

Action

- To display VCCP statistics for both member routers in the Virtual Chassis:

  `user@host>` **show virtual-chassis protocol statistics all-members**

- To display VCCP statistics for a specified member router 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 on which you are issuing the command:

  `user@host>` **show virtual-chassis protocol statistics local**

Related Documentation

- Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 314
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

CHAPTER 27

MX Series Virtual Chassis Configuration Statements

This chapter provides a reference for each of the MX Series Virtual Chassis configuration statements. The statements are organized alphabetically.

member (MX Series Virtual Chassis)

Syntax

```
member member-id {
  role (routing-engine | line-card);
  serial-number serial-number;
}
```

Hierarchy Level

[edit virtual-chassis]

Release Information

Statement introduced in Junos OS Release 11.2.

Description

Configure an MX Series router as a member of a Virtual Chassis configuration. You can configure a maximum of two member routers in an MX Series Virtual Chassis.

Options

member-id—Numeric value that identifies a member router in a Virtual Chassis configuration.

Values: 0 or 1

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

no-split-detection (MX Series Virtual Chassis)

Syntax no-split-detection;

Hierarchy Level [edit virtual-chassis]

Release Information Statement introduced in Junos OS Release 11.2.

Description As part of the preprovisioned configuration for an MX Series Virtual Chassis, disable detection of a split in the Virtual Chassis configuration. By default, split detection in the Virtual Chassis is enabled. To maintain the Virtual Chassis configuration in the event of a failure of one of the two member routers, we recommend that you use the **no-split-detection** statement to disable split detection in Virtual Chassis configurations in which you think the backup router is more likely to fail than the link to the backup router.

BEST PRACTICE: **We recommend that you use the no-split-detection statement to disable split detection for a two-member MX Series Virtual Chassis configuration if you think the backup router is more likely to fail than the Virtual Chassis port links to the backup router. Configuring redundant Virtual Chassis ports on different line cards in each member router reduces the likelihood that all Virtual Chassis port interfaces to the backup router can fail.**

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Disabling Split Detection in a Virtual Chassis Configuration on page 335
- Split Detection Behavior in a Virtual Chassis on page 301
- Global Roles and Local Roles in a Virtual Chassis on page 294
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

preprovisioned (MX Series Virtual Chassis)

Syntax preprovisioned;

Hierarchy Level [edit virtual-chassis]

Release Information Statement introduced in Junos OS Release 11.2.

Description Enable creation of a Virtual Chassis by means of a preprovisioned configuration.

To configure a Virtual Chassis consisting of MX Series routers, you must create a preprovisioned configuration on the master router in the Virtual Chassis by specifying the serial number, member ID, and role for each router (member chassis) in the Virtual Chassis. When a new member router joins the Virtual Chassis, its serial number is compared against the values specified in the preprovisioned configuration. If the serial number of a joining router does not match any of the configured serial numbers, the software prevents that router from becoming a member of the Virtual Chassis.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

role (MX Series Virtual Chassis)

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

Hierarchy Level [edit virtual-chassis member *member-id*]

Release Information Statement introduced in Junos OS Release 11.2.

Description As part of the preprovisioned configuration for an MX Series Virtual Chassis, assign the role to be performed by each member router in the Virtual Chassis. The preprovisioned configuration permanently associates the member ID and role with the member router's chassis serial number.

Options **routing-engine**—Enable the member router to function as the master router or backup router of the Virtual Chassis configuration. The master router maintains the global configuration and state information for both members of the Virtual Chassis, and runs the chassis management processes. The backup router synchronizes with the master router and relays chassis control information, such as line-card presence and alarms, to the master router. If the master router is unavailable, the backup router takes mastership of the Virtual Chassis to preserve routing information and maintain network connectivity without disruption. You must assign the **routing-engine** role to both members of the Virtual Chassis. When the Virtual Chassis is formed, the software runs a mastership election algorithm to determine which of the two member routers functions as the master router and which functions as the backup router of the Virtual Chassis.

line-card—Explicitly configuring a member router with the **line-card** role is *not supported* in the current release. However, when split detection is enabled (the default behavior for a Virtual Chassis) and either the Virtual Chassis ports go down or the backup router fails, the master router takes a **line-card** role. The **line-card** role effectively removes the former master router from the Virtual Chassis configuration until connectivity is restored.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345
- Virtual Chassis Components Overview on page 288
- Global Roles and Local Roles in a Virtual Chassis on page 294
- Split Detection Behavior in a Virtual Chassis on page 301

serial-number (MX Series Virtual Chassis)

Syntax serial-number *serial-number*;

Hierarchy Level [edit virtual-chassis member *member-id*]

Release Information Statement introduced in Junos OS Release 11.2.

Description As part of the preprovisioned configuration for an MX Series Virtual Chassis, specify the chassis serial number of each MX Series member router in the Virtual Chassis configuration. If you do not correctly specify a router's serial number in the preprovisioned configuration, the software does not recognize that router as a member of the Virtual Chassis.

Options ***serial-number***—Alphanumeric string that represents the chassis serial number of each member router in the Virtual Chassis configuration. The chassis serial number is located on a label affixed to the side of the MX Series chassis. You can also obtain the router's chassis serial number by issuing the **show chassis hardware** command.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

traceoptions (MX Series Virtual Chassis)

Syntax

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

Hierarchy Level [edit virtual-chassis]

Release Information Statement introduced in Junos OS Release 11.2.

Description Define tracing operations for the MX Series Virtual Chassis configuration.

Default Tracing operations are disabled.

Options **detail**—(Optional) Generate detailed trace information for a flag.

disable—(Optional) Disable a flag.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 3 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **all**—All tracing operations.
- **auto-configuration**—Trace Virtual Chassis ports that have been automatically configured.
- **csn**—Trace Virtual Chassis complete sequence number (CSN) packets.
- **error**—Trace Virtual Chassis errored packets.
- **graceful-restart**—Trace Virtual Chassis graceful restart events.
- **hello**—Trace Virtual Chassis hello packets.
- **krt**—Trace Virtual Chassis kernel routing table (KRT) events.
- **lsp**—Trace Virtual Chassis link-state packets.
- **lsp-generation**—Trace Virtual Chassis link-state packet generation.
- **me**—Trace Virtual Chassis mastership election (ME) events.

- **normal**—Trace normal events.
- **packets**—Trace Virtual Chassis packets.
- **parse**—Trace reading of the configuration.
- **psn**—Trace partial sequence number (PSN) packets.
- **route**—Trace Virtual Chassis routing information.
- **spf**—Trace Virtual Chassis shortest-path-first (SPF) events.
- **state**—Trace Virtual Chassis state transitions.
- **task**—Trace Virtual Chassis task operations.

match *regular-expression*—(Optional) Refine the output to include lines that contain the regular expression.

no-stamp—(Optional) Do not place a timestamp on any trace file.

no-world-readable—(Optional) Restrict file access to the user who created the file.

receive—(Optional) Trace received packets.

replace—(Optional) Replace a trace file instead of appending information to it.

send—(Optional) Trace transmitted packets.

size *maximum-file-size*—(Optional) Maximum size of each trace file. By default, the number entered is treated as bytes. Alternatively, you can include a suffix to the number to indicate kilobytes (KB), megabytes (MB), or gigabytes (GB). If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: ***size*k** to specify KB, ***size*m** to specify MB, or ***size*g** to specify GB

Range: 10240 through 1073741824

world-readable—(Optional) Enable unrestricted file access.

Required Privilege Level

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

Related Documentation

- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Tracing Virtual Chassis Operations for MX Series 3D Universal Edge Routers on page 337
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

virtual-chassis (MX Series Virtual Chassis)

Syntax

```
virtual-chassis {
  member member-id {
    role (routing-engine | line-card);
    serial-number serial-number;
  }
  no-split-detection;
  preprovisioned;
  traceoptions {
    file filename <files number> <match regular-expression> <no-stamp> <replace> <size
      maximum-file-size> <world-readable | no-world-readable>;
    flag flag <detail> <disable> <receive> <send>;
  }
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 11.2.

Description Create a Virtual Chassis configuration for MX Series routers.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Preprovisioned Member Information for a Virtual Chassis on page 321
- Example: Configuring Interchassis Redundancy for MX Series 3D Universal Edge Routers Using a Virtual Chassis on page 345

PART 10

Index

Index

Symbols

A

B

C

D

H

I

K

L

M

N

Index of Statements and Commands

S

T

V

