

Junos® OS

Junos® OS Software Installation and Upgrade Guide

Published
2025-07-15

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

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

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

Junos® OS Junos® OS Software Installation and Upgrade Guide
Copyright © 2025 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

[About This Guide | xvi](#)

1

[Junos OS Overview](#)

[Junos OS Overview | 2](#)

[Junos OS Overview | 2](#)

[Junos OS Overview | 2](#)

2

[Install, Upgrade, or Downgrade for Junos OS Software](#)

[Junos OS Install and Upgrade Overview | 8](#)

[Junos OS Installation and Upgrade Overview | 8](#)

[How You Can Get Started | 9](#)

[Best Practices for Upgrading Junos OS | 15](#)

[Junos OS Installation Package Names | 15](#)

[Junos OS Installation Packages Prefixes | 17](#)

[Junos OS Release Numbers | 22](#)

[Junos OS Editions | 23](#)

[Boot Sequence on Devices with Routing Engines \(Junos OS\) | 24](#)

[Storage Media and Routing Engines \(Junos OS\) | 27](#)

[Routing Engines and Storage Media \(Junos OS\) | 27](#)

[Repartitioning Routing Engine System Storage to Increase the Swap Partition \(Junos OS\) | 28](#)

[System Memory and Storage Media on Routers \(Junos OS\) | 29](#)

[Routing Engines and Storage Media Names \(ACX Series, M Series, MX Series, PTX Series, T Series, TX Matrix, TX Matrix Plus, and JCS 1200 Routers\) | 32](#)

[System Memory and Storage Media for SRX Series Firewalls | 35](#)

[Accessing USB Storage on PTX1000 Routers | 44](#)

[Prepare for Installation and Upgrade | 47](#)

[Preparing for Software Installation and Upgrade \(Junos OS\) | 47](#)

[Upgrade or Reinstall Junos OS | 48](#)

[Checklist for Reinstalling Junos OS | 48](#)

[Log the Software Version Information \(Junos OS\) | 50](#)

[Log the Hardware Version Information \(Junos OS\) | 52](#)

Log the Chassis Environment Information (Junos OS)	54
Log the System Boot-Message Information (Junos OS)	55
Log the Active Configuration (Junos OS)	57
Log the Interfaces on the Router (Junos OS)	58
Log the BGP, IS-IS, and OSPF Adjacency Information (Junos OS)	60
Log the System Storage Information (Junos OS)	62
Validating the Configuration Image Before Upgrading or Downgrading the Software (Junos OS)	63
Ensure Sufficient Disk Space for Junos OS Upgrades on SRX Series Firewalls	64
Verify Available Disk Space on SRX Series Devices	64
Cleanup the System File Storage Space	65
Verifying Junos OS and Boot Loader Software Versions on an EX Series Switch	66
Verifying the Number of Partitions and File System Mountings	67
Verifying the Loader Software Version	68
Verifying Which Root Partition Is Active	70
Verifying the Junos OS Version in Each Root Partition	71
Access Juniper Support	72
Existing Users—How to Log In	72
New Users—How to Create an Account	72
Downloading Software (Junos OS)	73
Downloading Software Using a Browser (Junos OS)	73
Downloading Software Using the Command-Line Interface (Junos OS)	75
Download Software Using Download Manager (SRX Series Only)	77
Reinstall Junos OS	79
Reconfigure Junos OS	80
Configure Host Names, Domain Names, and IP Addresses (Junos OS)	80
Protect Network Security by Configuring the Root Password	81
Check Network Connectivity (Junos OS)	83
Managing Disk Space for Junos OS Installation	84
Verifying PIC Combinations (Junos OS)	85
Upgrade to Junos OS with Upgraded FreeBSD 	87
Upgrade and Downgrade for Junos OS with Upgraded FreeBSD	87
Before You Upgrade, Install os-package	87
Install Junos OS with Upgraded FreeBSD Over Junos OS with Upgraded FreeBSD of a Different Release	89
Upgrade to a Junos OS Release with Upgraded FreeBSD	91

- Determine Which Package or Packages to Install | **92**

- Install a Junos OS with Upgraded FreeBSD Release Over an Existing Junos OS Release | **94**

Downgrading from Junos OS with Upgraded FreeBSD | **97**

- Downgrading from Junos OS with Upgraded FreeBSD to Legacy Junos OS | **98**

- Downgrading from Junos OS with Upgraded FreeBSD Release 17.4 or Later to Release 15.1 Through 17.3 | **99**

- Downgrading from Junos OS with Upgraded FreeBSD Release 17.3 or Earlier to Release 15.1 Through 17.2 | **100**

- Downgrading from Junos OS with Upgraded FreeBSD Release 18.1 or Later to Release 17.4 or Later | **101**

Install Software on Routers | 102

Installing Software on Routing Devices (Junos OS) | **102**

- Installing the Software Package on a Router with a Single Routing Engine (Junos OS) | **102**

- Installing the Software Package on a Device with Redundant Routing Engines (Junos OS) | **104**

- Preparing the Device for the Installation (Junos OS) | **104**

- Installing Software on the Backup Routing Engine (Junos OS) | **106**

- Installing Software on the Remaining Routing Engine (Junos OS) | **108**

- Finalizing the Installation (Junos OS) | **110**

Installing Software on ACX Series Routers (Junos OS) | **113**

- Installing Junos OS Using a USB Storage Device on ACX Series Routers | **113**

- Installing Junos OS Upgrades from a Remote Server on ACX Series Routers | **117**

Configuring Root Partitions on ACX Series Routers | **118**

- Dual-Root Partitioning ACX Series Routers Overview | **118**

- Understanding How the Primary Junos OS Image with Dual-Root Partitioning Recovers on the ACX Series Router | **120**

- Junos OS Release 12.2 or Later Upgrades with Dual-Root Partitioning on ACX Series Routers | **121**

- Example: Installing Junos OS and Configuring a Dual-Root Partition on ACX Series Routers Using the CLI | **122**

- Requirements | **122**

- Overview | **122**

- Configuration | **123**

- Verification | **126**

Installing Software on MX Series Routers Using a USB Flash Drive | **127**

- Pre-Installing Junos OS on a USB Flash Drive | **128**

- Installing Junos OS from a USB Flash Drive | **128**

Upgrading Junos OS using a USB Flash Drive | 129

Installing Software on Routing Devices (Junos OS) | 132

Installing the Software Package on a Router with a Single Routing Engine (Junos OS) | 132

Installing the Software Package on a Device with Redundant Routing Engines (Junos OS) | 134

Install Software on Switches | 143

Installing Software on EX Series Switches | 143

Understanding Software Installation on EX Series Switches | 144

Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine (CLI Procedure) | 146

Installing Software on an EX Series Switch with Redundant Routing Engines (CLI Procedure) | 149

Preparing the Switch for the Software Installation | 150

Installing Software on the Backup Routing Engine | 152

Installing Software on the Default Primary Routing Engine | 153

Returning Routing Control to the Default Primary Routing Engine (Optional) | 155

Upgrading the Loader Software on the Line Cards in a Standalone EX8200 Switch or an EX8200 Virtual Chassis | 156

Booting an EX Series Switch Using a Software Package Stored on a USB Flash Drive | 160

Configuring Dual-Root Partitions | 161

Troubleshooting Software Installation on EX Series Switches | 165

Recovering from a Failed Software Upgrade on an EX Series Switch | 165

Rebooting from the Inactive Partition | 167

Freeing Disk Space for Software Installation | 168

Installation from the Boot Loader Generates 'cannot open package' Error | 169

Troubleshooting a Switch That Has Booted from the Backup Junos OS Image | 170

Boot EX4000 using Network Boot | 172

Installing Software on QFX Series Devices (Junos OS) | 174

Installing Software Packages on QFX Series Devices (Junos OS) | 174

Installing the Software on QFX10002-60C Switches | 175

Installing a Standard Software Package on QFX5000 and EX4600 Switches | 176

Installing a Standard Software Package on QFX10002 Switches | 177

Installing a Software Package on QFX10008 and QFX10016 Switches | 178

Upgrading Software by Using Automatic Software Download for Switches (Junos OS) | 182

Configuring DHCP Services for the Switch (Junos OS) | 183

Enabling Automatic Software Download on a Switch (Junos OS) | 183

Verifying That Automatic Software Download Is Working Correctly (Junos OS) | 184

Installing Junos OS Software with Junos Automation Enhancements | 185

Installing and Recovering Software Using the Open Network Install Environment (ONIE) | 191

Understanding the Open Network Install Environment | 192

Downloading Software Files with a Browser | 193

Connecting to the Console Port | 193

Backing Up the Current Configuration Files | 194

Uninstalling the Existing Version of Junos OS | 194

Installing a Junos OS Software Package That Resides on a Webserver or DHCP Server with DHCP Options Configured | 195

Installing Junos OS Software Using Secure Copy Protocol (SCP) | 196

Installing Junos OS Software Using FTP or TFTP Without a Webserver | 197

Installing Junos OS Software Using DHCP Server with No DHCP Options Configured | 198

Installing Junos OS Software Using Webserver Without DHCP Configured | 199

Installing Junos OS Software Using USB Media | 200

Verifying Software Installation | 200

Troubleshooting Boot Problems | 201

Creating an Emergency Boot Device | 202

Performing a Recovery Installation | 203

Manage YANG Packages During an Upgrade or Downgrade | 205

Managing YANG Packages and Configurations During a Software Upgrade or Downgrade | 205

Backing up and Deleting the Configuration Data | 206

Restoring the YANG Packages and Configuration Data | 207

3

Install Software on Firewalls

Junos OS Upgrades for SRX Series Firewalls | 210

Upgrades for SRX Series Firewalls | 210

Upgrade Junos OS Package on SRX Series Firewalls | 211

Requirements | 212

Installation Overview | 213

Configuration | 213

Verify Junos OS Upgrade | 215

Install Junos OS on SRX Series Firewalls Using the Partition Option | 216

- Requirements | 216
- Installation Overview | 216
- Configuration | 217
- Verify the Partition Scheme Details | 219

Revert the Junos OS Software Image to the Previous Release | 220

- Requirements | 220
- Overview | 220
- Configuration | 221
- Verify Installation of Junos OS Downgrade Package | 222
- Platform-Specific Rollback Behavior | 222

Reboot and Halt SRX Series Firewall | 223

Reboot SRX Series Firewall | 223

- Requirements | 224
- Overview | 224
- Configuration | 224
- Verify the Firewall Reboot | 225

Halt the SRX Series Firewalls | 226

- Requirements | 226
- Overview | 226
- Configuration | 226
- Verify the Firewall Halt | 228

Platform-Specific Halting Software Processes Behavior | 228

Manage Chassis Component Status on SRX Series Firewalls: Online and Offline Operations | 229

- Restart the Chassis on SRX Series Firewalls | 230

Prepare the USB Flash Drive to Upgrade Junos OS on SRX300 Line of Firewall Devices | 231

USB Flash Drives Usage on SRX Series Firewall Overview | 231

- Prepare the USB Flash Drive | 232

Install Junos OS on SRX300 Line of Firewalls Using a USB Flash Drive | 233

- Install Junos OS on a Flash Drive | 234
- Platform-Specific USB Flash Drive Behavior | 236

Upgrade and Install the Boot Loader on SRX Series Firewalls | 237

Upgrade the Bootloader on SRX Series Firewalls | 237

Install Junos OS on SRX300 Line of Firewalls from the Boot Loader Using a TFTP Server | 239

Install Junos OS on SRX300 Line of Firewalls from the Boot Loader Using a USB Storage Device | 242

Upgrade the Software of SRX Series Firewalls by Using a PXE Boot Server | 243

Upgrade SRX1500 Firewall Software | 243

Upgrade SRX4200 Firewall Software | 247

Upgrade the Software of SRX4600 Device | 250

Configure Root Partitions on SRX Series Devices | 254

Dual-Root Partition on SRX Series Firewalls | 254

Reinstall Single-Root Partition on SRX Series Firewalls | 260

Platform-Specific Storage Media Behavior | 260

Veriexec and OpenPGP | 262

Veriexec File Signing and Verification for Junos OS | 262

Generate OpenPGP Keys for Secure Package Signing on Junos OS | 268

4

Install and Upgrade the BIOS and Firmware

For Routers | 273

Installing and Upgrading Firmware | 273

Before You Begin Installing or Upgrading the Firmware | 273

Installing Firmware on the 100-Gigabit DWDM OTN MIC (MIC3-100G-DWDM) | 275

Upgrading Firmware on the 100-Gigabit DWDM OTN MIC (MIC3-100G-DWDM) | 276

Installing Firmware on ACX6360 Router | 278

Upgrading Firmware on the ACX6360 Router | 279

For Switches | 283

Upgrading System CPLD, BIOS, CPU CPLD, PoE Firmware, and eMMC Firmware for EX4400 Devices | 283

Checking Latest Firmware Versions | 283

Upgrading BIOS | 284

Upgrading CPU CPLD | 285

Upgrading System CPLD | 286

Upgrading PoE Firmware | 288

Upgrading PoE Firmware Using jfirmware | 290

Upgrading eMMC Firmware | 293

Upgrading Firmware in Virtual Chassis | 294

Upgrading U-Boot, PoE Firmware, eUSB Firmware, and System CPLD for EX4100 Devices | 295

Upgrading U-Boot | 295

Upgrading System CPLD | 297

Upgrading PoE Firmware | 298

Upgrading PoE Firmware Using jfirmware | 298

Upgrading eUSB Firmware | 298

Upgrading Firmware in Virtual Chassis | 299

Upgrade Firmware on QFX5120-48Y and EX4650-48Y using the jfirmware Package | 300

For Firewalls | 304

Upgrade BIOS and Firmware (SRX Series Firewall Only) | 304

BIOS Upgrades on SRX Series Firewalls | 304

Disable Auto BIOS Upgrade on SRX Series Firewalls | 306

Upgrade 5.1KW HVAC/HVDC Single and Dual Input PSM Firmware (SRX5800) | 307

5

System Backup and Recovery

Back Up an Installation | 310

Backing Up an Installation Using Snapshots (Junos OS) | 310

Understanding How to Back Up an Installation on Switches | 310

Creating a Snapshot and Using It to Boot a QFX Series Switch | 312

Creating a Snapshot on an External USB Flash Drive and Using It to Boot a QFX Series Switch | 313

Creating a Snapshot and Using It to Boot a QFX3500 and QFX3600 Series Switch | 314

Creating a Snapshot and Using It to Boot an EX Series Switch | 317

Creating a Snapshot on a USB Flash Drive and Using It to Boot the Switch | 317

Creating a Snapshot and Using It to Boot an SRX Series Firewall | 318

Create a Snapshot and Using It to Boot an SRX Series Firewall | 318

Backup the Current Installation on SRX Series Firewalls | 320

Creating a Snapshot and Using It to Boot an ACX Series Router | 322

Understanding System Snapshot on an ACX Series Router | 322

Example: Taking a Snapshot of the Software and Configuration | 323

Recover Junos OS | 328

Recovery of Junos OS | 328

- Recovering from a Failed Software Installation | 328
- Recovering Junos OS on a Device Running Junos OS with Upgraded FreeBSD | 331

Recovery Using an Emergency Boot Device (Junos OS) | 334

- Creating an Emergency Boot Device for Routers | 335
- Creating an Emergency Boot Device for QFX Series Switches | 337
- Recovering the Installation Using an Emergency Boot Device on QFX Series Switches | 338
- Performing a Recovery Installation | 341

How to Recover Junos OS with Upgraded FreeBSD | 343

- Ways to Recover Junos OS with Upgraded FreeBSD Without the Use of the CLI | 343
- How to Access the Junos Main Menu, Boot Menu, and Options Menu | 347
 - How to Access the Junos Main Menu | 347
 - How to Access the Boot Menu | 349
 - How to Access the Options Menu | 350

Rescue and Recovery of Configuration File (Junos OS) | 351

- Saving and Reverting a Rescue Configuration File | 351
 - Saving a Rescue Configuration File | 351
 - Reverting to the Rescue Configuration | 355
- Copy Backup Configurations and Restore Saved Configurations | 356
 - Copy Backup Configurations to the Router | 356
 - Restoring a Saved Configuration | 357
- Reverting to the Default Factory Configuration by Using the request system zeroize Command | 359

Autorecovery of Configuration, Licenses, and Disk Information on SRX Series Devices | 360

Install, Upgrade, and Downgrade VM Host Software, and Back Up and Recover VM Host Devices

VM Host Overview and CLI Commands | 365

VM Host Overview (Junos OS) | 365

- What Are VM Hosts? | 365
- Routing Engines with VM Host Support | 367
- Salient Features of the Routing Engines with VM Host Support | 372
- Enhanced VM Host Architecture | 379

VM Host Operations and Management | 380

Boot Process for Routers with VM Host Support | 382

Boot Process for Routers with VM Host Support | 382

Install, Upgrade, Back Up, and Recover VM Host | 385

Installing, Upgrading, Backing Up, and Recovery of VM Host | 385

Copy the VM Host Installation Package to the PXE Boot Server | 392

Copying VM Host Installation Package to the PXE Boot Server | 392

Upgrade NFX Devices | 396

Upgrade the NFX250 Software to NFX250 NextGen Software | 396

NFX250 NextGen Software Upgrade Overview | 396

Prerequisites | 397

Upgrade to NFX250 NextGen Software Architecture | 399

Upgrading the Junos OS on NFX Devices | 399

Upgrade Dual-Disk Partitions on NFX250 NextGen and NFX350 Devices | 404

Downgrade Instructions for NFX Series Devices Running Junos OS Release 23.1R1 | 415

Create an Emergency Boot Device for Routing Engines with VM Host Support | 416

Creating an Emergency Boot Device for Routing Engines with VM Host Support | 416

Upgrade Firmware on Routing Engines with VM Host Support | 418

Upgrading the SSD Firmware on Routing Engines with VM Host Support | 418

Upgrading the i40e NVM Firmware on Routing Engines with VM Host Support | 421

Disable Autorecovery on Routing Engines with VM Host Support | 433

Disabling Autorecovery on Routing Engines with VM Host Support | 433

Personality Upgrade | 434

Personality Upgrade Process | 434

Understanding the Personality Upgrade Process for a Device | 434

Supported Personality Upgrades | 436

Upgrading the Personality of a Device by Using a USB Flash Drive | 437

Upgrading the Personality of a Device by Using CLI | 438

How to Upgrade the Personality of a Device on Junos OS | 438

Upgrading the Personality of a Device by Using a PXE Boot Server | 441

Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices | 446

- Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the PXE Boot Server | **446**
- Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the USB Option | **450**
- Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the CLI Option | **451**
- Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using Zero Touch Provisioning (ZTP) | **454**

Zero Touch Provisioning and Secure Zero Touch Provisioning

Configure Zero Touch Provisioning and Secure Zero Touch Provisioning | 463

Zero Touch Provisioning | 463

- Zero Touch Provisioning Overview | **463**
- Zero Touch Provisioning Using DHCP Options | **468**
- Zero Touch Provisioning Using DHCPv6 Options | **489**
- Zero Touch Provisioning on SRX Series Firewalls | **496**
 - Understanding Zero Touch Provisioning on SRX Series Firewalls | **496**
 - Configuring Zero-Touch Provisioning on an SRX Series Firewall | **501**
 - Understanding Factory-Default Configuration on SRX Series Firewall for Zero Touch Provisioning | **505**

Monitoring Zero Touch Provisioning | 506

- Using the Console to Monitor Zero Touch Provisioning in Junos OS | **506**
- Using System Log Alerts to Monitor Zero Touch Provisioning | **507**
- Using Error Messages to Monitor Zero Touch Provisioning | **508**
- Using System Log Files to Monitor Zero Touch Provisioning in Junos OS Using DHCP Options | **508**
- Using System Log Files to Monitor Zero Touch Provisioning in Junos OS Using DHCPv6 Options | **510**
- Using the show dhcp client binding Command | **511**
- Using the show dhcpv6 client binding Command | **512**
- Using the show dhcp client statistics Command | **513**
- Using the show dhcpv6 client statistics Command | **514**

Secure Zero Touch Provisioning | 519

Switching between Secure Zero Touch Provisioning and Zero Touch Provisioning | 530

Phone-Home Client

Configure Phone-Home Client | 534

Obtaining Configurations and Software Image Without User Intervention Using Phone-Home Client | 534

Deploying the Phone-home Client and Zero Touch Provisioning on vSRX Virtual Firewall | 538

Provision a Virtual Chassis Using the Phone-Home Client | 541

Overview of Phone-Home Provisioning for a Virtual Chassis | 542

How To Enable Phone-Home Provisioning on a Virtual Chassis | 544

Phone-Home Process on a Virtual Chassis | 545

Startup and Request Provisioning Information from PHS | 546

Bootstrap Virtual Chassis Members | 548

Apply Scripts and New Configuration on the Virtual Chassis | 549

Provisioning Process Completion | 550

Phone-Home Provisioning Status Notifications | 550

Verify Virtual Chassis Status After Phone-Home Provisioning | 552

Troubleshoot Phone-Home Provisioning Issues | 553

9

Automatic Installation of Configuration Files

Configure Automatic Installation of Configuration Files | 556

Understanding Autoinstallation of Configuration Files (Junos OS) | 556

Autoinstallation Overview | 556

Autoinstallation Process on Satellite Devices in a Junos Node Unifier Group | 564

Configuring Autoinstallation of Configuration Files (Junos OS) | 567

Configuring Autoinstallation of Configuration Files (CLI Procedure) | 568

Example: Configure Autoinstallation on SRX Series Devices | 570

Requirements | 570

Overview | 571

Configuration | 571

Verification | 573

Verifying Autoinstallation Status | 574

Autoinstalling a Configuration File from a Disk-on-Key USB Memory Stick onto an EX2200 or EX3300 Switch | 575

Configuring Autoinstallation on JNU Satellite Devices | 578

Verifying Autoinstallation on JNU Satellite Devices | 581

Configuring Autoinstallation of Configuration Files on ACX Series (Junos OS) | 582

ACX Series Autoinstallation Overview | 583

Before You Begin Autoinstallation on an ACX Series Universal Metro Router	585
Autoinstallation Configuration of ACX Series Universal Metro Routers	586
Verifying Autoinstallation on ACX Series Universal Metro Routers	587
USB Autoinstallation on ACX Series Routers	588
Autoinstallation on ACX Series Routers in Hybrid Mode Overview	589
Prerequisites for Autoinstallation on ACX Series Routers in Hybrid Mode	590
Autoinstallation Process on a New ACX Series Router in Hybrid Mode	591
Configuring Autoinstallation of ACX Series Routers in Hybrid Mode	594

Configuration Statements and Operational Commands

Configuration Statements and Operational Commands	598
---	-----

Junos CLI Reference Overview	598
------------------------------	-----

About This Guide

Use this guide for information relevant to upgrading Junos OS and related software: software packages, upgrading and downgrading Junos OS releases, system backup and recovery procedures, installing and upgrading firmware, storage media, and ZTP.

1

PART

Junos OS Overview

- [Junos OS Overview | 2](#)
-

CHAPTER 1

Junos OS Overview

IN THIS CHAPTER

- [Junos OS Overview | 2](#)

Junos OS Overview

IN THIS SECTION

- [Junos OS Overview | 2](#)

Junos OS is the single operating system that powers Juniper's broad portfolio of physical and virtual networking and security products.

Junos OS Overview

IN THIS SECTION

- [One Operating System | 3](#)
- [One Modular Software Architecture | 3](#)
- [Secure Boot and Bootloader | 4](#)
- [Hardware Root of Trust | 5](#)
- [FIPS 140-2 Security Compliance | 6](#)

Juniper Networks provides high-performance network devices that create a responsive and trusted environment for accelerating the deployment of services and applications over a single network. The Junos® operating system (Junos OS) is the foundation of these high-performance networks.

Junos OS includes the following architecture variations:

- Junos OS FreeBSD 6 on bare metal. This is Junos OS based on a FreeBSD 6 kernel.
- Junos OS FreeBSD 10 or later on bare metal. This is Junos OS based on an upgraded FreeBSD kernel. Starting with Junos OS Release 15.1, certain hardware platforms run Junos OS with upgraded FreeBSD. Starting in Junos OS Release 16.1, Junos OS with upgraded FreeBSD can run as a guest virtual machine (VM) on a Linux VM host. For more on which platforms run Junos OS with upgraded FreeBSD, search for **Junos kernel upgrade to FreeBSD 10+** in Feature Explorer: [Junos kernel upgrade to FreeBSD 10+](#).
- Junos OS Evolved. See [Introducing Junos® OS Evolved](#) and the [Junos® OS Evolved Software Installation and Upgrade Guide](#) for more information about Junos OS Evolved.

Unlike other complex, monolithic software architectures, Junos OS incorporates key design and developmental differences to deliver increased network availability, operational efficiency, and flexibility. The following are key advantages to this approach:

One Operating System

Unlike other network operating systems that share a common name but splinter into many different programs, Junos OS is a single, cohesive operating system that is shared across all network devices and product lines. This allows Juniper Networks engineers to develop software features once and share these features across all product lines simultaneously. Because features are common to a single source, they generally are implemented the same way for all product lines, thus reducing the training required to learn different tools and methods for each product. Because all Juniper Networks products use the same code base, interoperability between products is not an issue.

One Modular Software Architecture

Although individual modules of Junos OS communicate through well-defined interfaces, each module runs in its own protected memory space, preventing one module from disrupting another. This separation enables the independent restart of each module as necessary. This is in contrast to monolithic operating systems where a malfunction in one module can ripple to other modules and cause a full system crash or restart. This modular architecture then provides for high performance, high availability, security, and device scalability not found in other operating systems.

The Junos OS is preinstalled on your Juniper Networks device when you receive it from the factory. Thus, when you first power on the device, all software starts automatically. You simply need to configure the software so that the device can participate in the network.

You can upgrade the device software as new features are added or software problems are fixed. You normally obtain new software by downloading the software installation packages from the Juniper Networks Support Web page onto your device or onto another system on your local network. You then install the software upgrade onto the device.

Juniper Networks routing platforms run only binaries supplied by Juniper Networks, and currently do not support third-party binaries. Each Junos OS image includes a digitally signed manifest of executables that are registered with the system only if the signature can be validated. Junos OS will not execute any binary without a registered signature. This feature protects the system against unauthorized software and activity that might compromise the integrity of your device.

Secure Boot and Bootloader

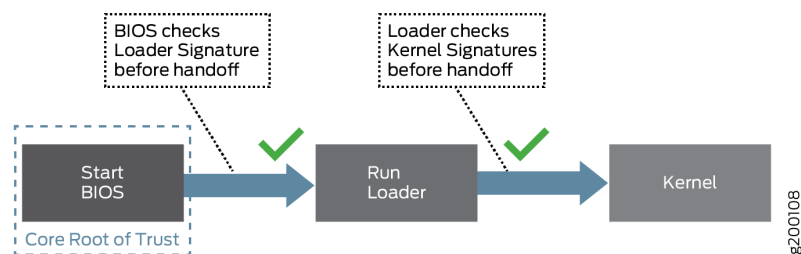
The system's boot-up process involves several stages, starting with the activation of the CPU as the boot processor, communicating with the PCH to bring up the system. The CPU jumps to the BIOS routine stored in the Primary SPI flash. In case of Primary flash failure, the boot FPGA switches to the Secondary flash for recovery, but it does not load the bootloader.

The bootloader plays a critical role in the system's boot-up process, ensuring secure and orderly execution of the operating system. The boot order and mechanisms like Secure Boot, USB, SSD, and PXE boot provide flexibility and resilience to the system's startup procedure. Additionally, GRUB configurations allow users to customize and troubleshoot the boot process as needed.

Secure Boot is a significant system security enhancement based on the UEFI standard (see www.uefi.org). It works by safeguarding the BIOS itself from tampering or modification and then maintaining that protection throughout the boot process.

The Secure Boot process begins with Secure Flash, which ensures that unauthorized changes cannot be made to the firmware. Authorized releases of Junos OS carry a digital signature produced by either Juniper Networks directly or one of its authorized partners. At each point of the boot-up process, each component verifies the next link is sound by checking the signature to ensure that the binaries have not been modified. The boot process cannot continue unless the signature is correct. This "chain of trust" continues until the operating system takes control. In this way, overall system security is enhanced, increasing resistance to some firmware-based persistent threats.

Figure 1 shows a simplified version of this "chain of trust."



Secure Boot requires no actions on your part to implement. It is implemented on supported hardware by default.

Secure Boot for SRX devices serves as a critical security mechanism designed to protect Juniper hardware and thwart the execution of unauthorized code or data. Unauthorized entities are defined as those lacking a proper Juniper digital signature or that of its authorized affiliates.

Here's a concise breakdown of how Secure Boot operates specifically for SRX1600, SRX2300, SRX4120, SRX4300 and SRX4700 devices:

- **Power-Up Sequence:** Upon powering up, the CPU initiates execution with the UEFI BIOS.
- **Boot Loader:** The UEFI BIOS loads the signed PE/COFF32+ executable `\EFI\BOOT\BOOTX64.EFI`. This executable contains a modified GRUB2, which allows only file read I/O to files with proper detached signatures.
- **Detached Signatures:** GRUB2 detached signatures are stored as `${object}.psig`, utilizing a binary (non-armored) OpenPGP format.
- **GPG Key Storage:** Additional GPG Keys, in conjunction with the trusted GPG public key compiled into GRUB2, may be stored in `\EFI\BOOT\grub-trusted.gpg`.
- **Trust Establishment:** Trust in `\EFI\BOOT\grub-trusted.gpg` is established by verifying `\EFI\BOOT\grub-trusted.gpg.psig` using the grub-root key embedded in `\EFI\BOOT\BOOTX64.EFI`.
- **GRUB2 Configuration:** The GRUB2 startup configuration, located in `\EFI\BOOT\grub-startup.cfg`, is responsible for loading the signed Linux kernel and, optionally, a signed initrd image.
- **File Loading Restrictions:** `BOOTX64.EFI` consistently insists on properly signed files and refrains from loading any unsigned files, excluding GRUB config or GRUB environment files.

Installation and Activation

To enable Secure Boot for SRX2300 and SRX4120 devices, navigate to the BIOS menu and select **Restore Secure Boot to Factory Settings** under the **Administer Secure Boot** menu.



NOTE: Once Secure Boot is enabled, it cannot be disabled.

For information on which Junos OS releases and hardware support Secure Boot, see [Feature Explorer](#) and enter **Secure Boot**.

Hardware Root of Trust

Hardware root of trust (HrOT) is a hardware-based security feature integrated in the system, which acts as a trusted foundation to verify the integrity of the firmware and ensure its secure operation. This feature provides an unalterable root of trust starting from the hardware, protecting against potential

security vulnerabilities within the system. HRoT acts as a critical component for guaranteeing the authenticity of the system firmware and configuration.

Unlike software-based trust mechanisms, HRoT is implemented directly in the hardware, making it highly resistant to tampering. The primary function of HRoT is to verify the integrity of the firmware, ensuring it hasn't been compromised or modified unauthorizedly. This feature is used to implement a secure boot process where only verified and trusted firmware can be loaded.



NOTE: Ensure the following message is displayed during the early bootstrap of the device to know that HRoT and secure boot are enforced.

JUNIPER HARDWARE ROOT OF TRUST WITH SECURE BOOT ENFORCED

FIPS 140-2 Security Compliance

For advanced network security, a special version of Junos OS, called Junos-FIPS 140-2, is available. Junos-FIPS 140-2 provides customers with software tools to configure a network of Juniper Networks devices in a FIPS environment. FIPS support includes:

- Upgrade package to convert Junos OS to Junos-FIPS 140-2
- Revised installation and configuration procedures
- Enforced security for remote access
- FIPS user roles (Crypto Officer, User, and Maintenance)
- FIPS-specific system logging and error messages
- IPsec configuration for Routing Engine-to-Routing Engine communication
- Enhanced password creation and encryption

Starting in Junos OS Release 15.1, Junos-FIPS is packaged in a domestic image only: a single Junos OS image supports both domestic and FIPS features. Users that have the FIPS credentials and permission to login can flip between a regular Junos image and FIPS image.



NOTE: Junos-FIPS has special password requirements. FIPS passwords must be between 10 and 20 characters in length. Passwords must use at least three of the five defined character sets (uppercase letters, lowercase letters, digits, punctuation marks, and other special characters). If Junos-FIPS is installed on the device, you cannot configure passwords unless they meet this standard.

2

PART

Install, Upgrade, or Downgrade for Junos OS Software

- [Junos OS Install and Upgrade Overview | 8](#)
 - [Prepare for Installation and Upgrade | 47](#)
 - [Upgrade to Junos OS with Upgraded FreeBSD | 87](#)
 - [Install Software on Routers | 102](#)
 - [Install Software on Switches | 143](#)
 - [Manage YANG Packages During an Upgrade or Downgrade | 205](#)
-

Junos OS Install and Upgrade Overview

IN THIS CHAPTER

- [Junos OS Installation and Upgrade Overview | 8](#)
- [Storage Media and Routing Engines \(Junos OS\) | 27](#)

Junos OS Installation and Upgrade Overview

SUMMARY

You receive a Juniper Networks device with the Junos® OS preinstalled. When you power on the device, it uses the installed software to start (boot). As new features and software fixes become available, you must upgrade your software for better user experience. Before the upgrade, back up the configuration files. Read this topic to understand what you'll gain when you upgrade Junos OS.

IN THIS SECTION

- [How You Can Get Started | 9](#)
- [Best Practices for Upgrading Junos OS | 15](#)
- [Junos OS Installation Package Names | 15](#)
- [Boot Sequence on Devices with Routing Engines \(Junos OS\) | 24](#)

In the rapidly changing era of mobile, cloud, and the Internet of Things (IoT) technologies, network and security challenges are commonplace. A legacy OS (OS) for network infrastructure is insufficient to address these challenges. An outdated version of software on your devices increases risks, such as cyber attacks, to both your users and the network environment. In addition, the complexity of maintaining an outdated OS can lower your team's operational efficiency and cost valuable resources such as time and money. You also run the risk of incurring business loss due to noncompliance with government and other organizational regulations because of an outdated OS on your devices.

We understand that you might have concerns about upgrading to the latest Junos OS, including:

- Network downtime and maintenance disrupting business continuity
- Higher operational cost and lower employee productivity because of learning curves and training
- Configuration compatibility between releases

However, the benefits of upgrading to the latest supported Junos OS often outweigh the potential risks of using an outdated OS. A newer version of the Junos OS includes new features, enhancements, and bug fixes; many customers find the value of upgrading to a new version beneficial to their organization with immediate returns from an upgrade. Here are the top benefits of keeping your software up to date:

Figure 1: Benefits of an Upgrade



Increased Efficiency and Compatibility

Improve IT operational efficiency with new features and bug fixes from previous versions. You can remove the complexity and additional cost of maintaining an older software version and free up time and resources for business innovation. You can also achieve better compatibility and integration with other devices in your network.



Customer Engagement

Deploy new services that will help to gain new customers or increase the loyalty of existing ones.



Business Growth and Compliance

Stay current with the latest technology and respond quickly and confidently to changing business needs. Comply with regulatory and audit requirements and avoid downtime by upgrading versions that are approaching end of life (EOL).



Better Security

Improve and simplify your deployments with software upgrades that include security patches.

jn-000843

How You Can Get Started

IN THIS SECTION

- [Considerations for Junos OS Installation](#) | 11

A Juniper Networks device is delivered with Junos OS preinstalled. When you power on the device, it starts (boots) using the installed software. As new features and software fixes become available, you must upgrade your software to use these features and fixes.

When you upgrade (or downgrade) a Junos OS version, you must first copy a software installation package to your device or other system on your local network. Then you use the CLI to install the new software on the device. You then reboot the device, which boots from the newly installed software.

The first step is to determine the required software version. For more information about software versions, see [Junos Software Versions - Suggested Releases to Consider and Evaluate](#).



NOTE: Before you install software on a device that is configured with custom YANG data models, back up and remove the configuration data corresponding to these data models from the active configuration. For more information, see "[Managing YANG Packages and Configurations During a Software Upgrade or Downgrade](#)" on page 205.

We understand that upgrading on an infrastructure device might require a scheduled downtime as well as tasks before and after the upgrade. Additionally, you might need to create plans and document the upgrade steps to ensure a successful outcome.

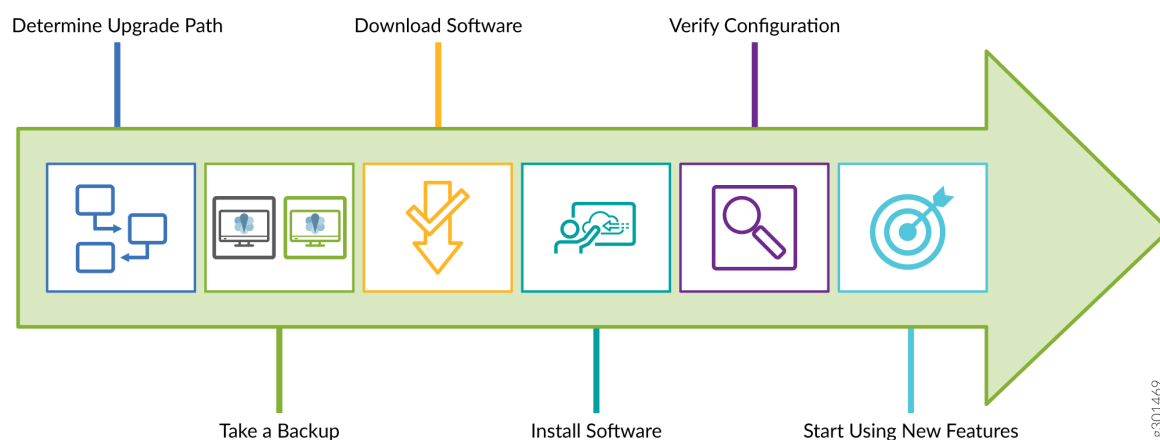
When you plan for an upgrade, we recommend that you review:

- Product Data Sheets accessible from [Products & Services](#) or contact your Juniper Account Team or your Juniper Partner.
- Knowledge Base (KB) articles that are related to Junos OS upgrade on our [Support](#) site.
- The [Juniper Licensing User Guide](#) to learn about Junos OS software licensing and which software licenses you might need for your device. The following [Table 1 on page 10](#) provides a quick reference taking you to the licensing information for your particular platform.

Table 1: Junos OS Licensing References

Junos OS Devices	Reference Topics
Juniper Networks® EX Series Switches	Software Licenses for EX Series Switches
Juniper Networks® MX Series Universal Routers	Software Licenses for MX Series Routers and MPC Service Cards
Juniper Networks® QFX Series Switches	Software Licenses for QFX Series Switches
Juniper Networks® SRX Series Firewalls	Software Licenses for SRX Series Firewalls

We provide a simple upgrade path that enables you to quickly and easily upgrade your Junos OS software and start using the new capabilities on your device. You can perform the upgrade as shown in the following illustration:



- To determine your upgrade path, see the KB article at [Junos Software Versions - Suggested Releases to Consider and Evaluate](#) and talk to support at [Contact Support](#).
- To take a backup, see ["Back Up the Current System's Files" on page 12](#).
- To download the software, see [Downloads](#). You can find the related KB articles at the [Support](#) site.
- To install and to verify the software, see:
 - [Upgrade and Downgrade for Junos OS with Upgraded FreeBSD](#)
 - [Installing, Upgrading, Backing Up, and Recovery of VM Host](#)
 - [Installing Software on Routing Devices \(Junos OS\)](#) for routers.
 - [Installing Software on Switches](#) for switches.
 - [Installing Software on Firewalls](#) for firewalls.
 - [request system software add \(Junos OS\)](#)

Considerations for Junos OS Installation

The following subsections introduce the overall considerations in installing the software:

Select the Junos OS Installation Type

The three types of installations that you can use to upgrade or downgrade the Junos OS running on your device are standard installation, category change installation, and recovery installation. Before you select the installation type, see ["Junos OS Installation Package Names" on page 15](#) to understand the different Junos OS installation packages. To understand which installation type best fits your need, see the following table.

Standard Installation

The standard installation type is the typical method you use to upgrade or downgrade software on the server. With this method, you can use an installation package that matches the installation package already installed on your device.

Category Change Installation

The category change installation type is the process you use to move from one edition of Junos OS to another on the same device. For example, moving from a Junos OS standard installation to a Junos-FIPS installation on a router.

When you move from one installation category to another, you must be aware of the restrictions regarding this change.

Juniper Networks does not support using the `request system software rollback` command to restore a different installation category on the device. When you install a different Junos OS category on a device, immediately after the installation is complete, you must run the `request system snapshot` command to delete the backup installation from the system.



NOTE: Starting in Junos OS Release 15.1R1, we've introduced a simplified edition scheme—*Junos* and *Junos Limited* editions. The *Junos* edition has full cryptography support. The *Junos Limited* edition has limited cryptography support. You cannot change between editions.

Recovery Installation

The recovery installation type is the process you use to repair a device with damaged software or a condition that prevents the upgrade, downgrade, or change in installation category of the software.

Back Up the Current System's Files

We recommend that you create a backup of your device before you start an upgrade. Your device can:

- Boot from a backup and come back online in case of an upgrade failure or corruption of the primary boot device because of a power failure during the upgrade.
- Retrieve the active configuration files and log files from the backup.
- Recover from a known, stable environment in case of an unsuccessful upgrade.

During a successful upgrade, the upgrade package completely reinstalls the existing OS. Although the upgrade process removes all stored files, it retains only the **juniper.conf** and SSH files. Other information is removed. The **juniper.conf** file stores the operational device configuration. Therefore, you must back up your existing configuration in case you need to return to it after running the installation program.

To take a backup, you use the system snapshot feature to take a snapshot of the files currently used to run the device. The snapshot captures the complete contents of the **/config** and **/var** directories, which

include the software that runs on your device, the active configuration, and the rescue configuration. The device copies these files into an alternative (an internal flash or an external USB flash) memory source. You can then use this snapshot to boot the device at the next boot up or as a back-up boot option. When you complete the backup, the existing and backup software installations are identical.



NOTE: Snapshots taken with the `request system snapshot` command in a system with Junos OS with upgraded FreeBSD are not the same as snapshots taken with the `request system snapshot` command in a legacy Junos OS system. To back up your Junos OS with upgraded FreeBSD system devices, use the `request system snapshot recovery` command.

To determine if your system uses Junos OS with upgraded FreeBSD, see: [Junos kernel upgrade to FreeBSD 10+](#).

When you issue the correct snapshot command, the `/root` file system is backed up to the `/altroot` file system and the `/config` file system is backed up to the `/altconfig` file system. The `/root` and `/config` file systems are on the device's flash memory device. The `/altroot` and `/altconfig` file systems are on the device's hard disk or solid-state drive (SSD). See [Backing Up an Installation Using Snapshots \(Junos OS\)](#) for more details.

We recommend that you understand the boot sequence on your device before you use the backup. See [Boot Sequence on Devices with Routing Engines \(Junos OS\)](#) for more details.

Determine Which Software Installation Package to Install

We deliver all software releases in signed packages that contain digital signatures to ensure that you download the official Juniper Networks software. You can either download software to the `/var/tmp` directory of your device, or install it directly from the Downloads page at <https://support.juniper.net/support/downloads/>.

To learn about the software packages currently running on a device, enter the `show version operational` mode command at the top level of the CLI. The output of the `show version` command displays the release number of the installed software but not the installed software edition.

For more information about signed software packages, see the ["Junos OS Installation Package Names" on page 15](#).

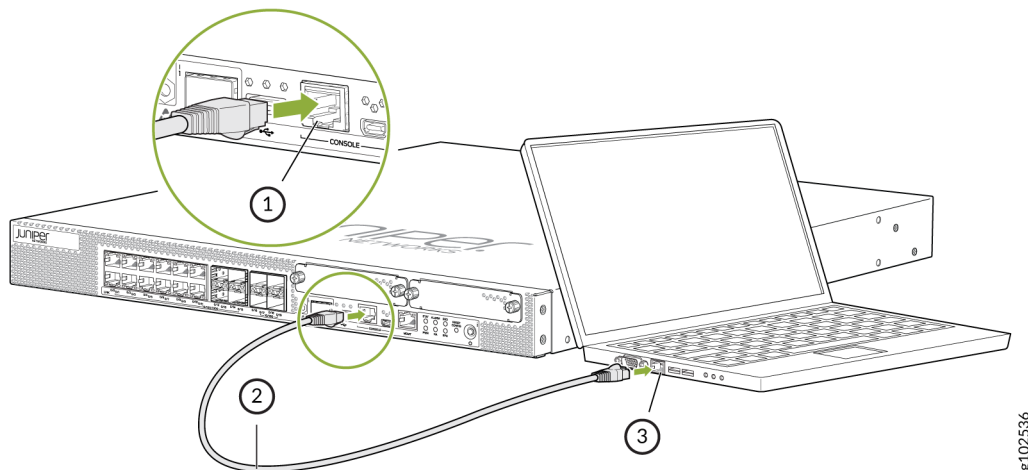
Upgrade Using an Out-of-Band Interface

We recommend that you upgrade all individual software packages using an out-of-band connection from the console or the management Ethernet interface. When you upgrade your Junos OS software by connecting to the console port on your Junos OS device, you can:

- Connect to the Junos OS device without using the network to which the device might or might not be connected. This connection creates a secondary path to the Junos OS device without relying on the network.
- Experience seamless connectivity (you might lose connectivity when you use in-band connections during an upgrade process).
- Have root access to the Junos OS device through a terminal or laptop interface, regardless of the state of the Junos OS device, unless it is completely powered off.
- Restore a Junos OS device or securely configure a minimum configuration through the terminal interface. A network operations technician located far away can perform these tasks securely on a terminal interface by using a modem, even if the primary network fails. Without a connection to the console port, a technician would have to visit the site to perform these tasks. A remote connection to the Junos OS device through a modem requires:
 - A cable and a connector, which are provided in the device accessory box.
 - A DB-9 to DB-25 (or similar) adapter for your modem, which you must purchase separately.

To configure the device initially, you must connect a terminal or laptop computer to the device through the console port, as shown in [Figure 2 on page 14](#).

Figure 2: Connect to the Console Port on a Juniper Networks Device



Validate the Installation Package with the Current Configuration

When you upgrade or downgrade software, we recommend that you include the `validate` option with the `request system software add` command to check whether the candidate software is compatible or not with

the current configuration. When you add a package with a different release number, Junos OS validates compatibility between the software and the current configuration by default.

Direct validation of the running configuration does not work when you upgrade to a release that is based on Junos OS with upgraded FreeBSD from a Junos OS release that is based on older versions of the FreeBSD kernel. Therefore, when you upgrade or downgrade between older Junos OS and Junos OS with upgraded FreeBSD, you might have to validate on a different host. For details, see [Upgrading and Downgrading to Junos with Upgraded FreeBSD](#).

If you do not want to validate during the upgrade, you must specify the `no-validate` option.

Best Practices for Upgrading Junos OS

We suggest that you start with the following best practices to optimize your upgrade experience:

- Read the [Junos OS Release Notes](#) for the release to which you are upgrading.
- Plan and document the upgrade steps to ensure a successful outcome.
- Connect your laptop or computer to the Junos OS device through the console port when you use the CLI to upgrade software (recommended).
- Connect your device to the Internet.
- Back up the current configuration and software.
- Delete or commit all uncommitted configuration changes.
- Clear files and erase unwanted or unused configurations using the `request system storage cleanup` command.
- (Only for Juniper Networks® SRX Series Firewalls) Ensure that both nodes of the SRX Series Firewall pair in a chassis cluster are online and have the same Junos OS version.
- Plan for an extended maintenance window, preferably during non-business hours, for the upgrade, troubleshooting, and any post-configuration procedures to minimize impact.
- Identify business contacts who will help verify application and network functionality after the upgrade.

Junos OS Installation Package Names

IN THIS SECTION

● [Junos OS Installation Packages Prefixes](#) | 17

- Junos OS Release Numbers | 22
- Junos OS Editions | 23

The installation package is used to upgrade or downgrade from one release to another. When installed, the installation package completely reinstalls the software, rebuilds the Junos OS file system, and can erase system logs and other auxiliary information from the previous installation. The installation package does, however, retain the configuration files from the previous installation.

A Junos OS installation package can have one of the following general patterns:

- ***prefix-platform-product-architecture-application-binary-interface-release-edition.extension*** (for installing with the `request system software add` command)
- ***prefix-media-media-keyword-platform-architecture-application-binary-interface-release-edition.extension*** (for images installed from the USB drive or the loader prompt)
- ***prefix-flex-release-edition.extension*** (for enhanced automation variants of Junos OS)

Table 2: Descriptions of Junos OS Package Name Fields

Field Name	Description
<i>prefix</i>	Package name prefix. Different products use different prefixes. These prefixes are explained later in this chapter.
host	Host is included in the package name when the platform is Linux based; this prefix indicates the image includes the host software as well as Junos OS.
media <i>media-keyword</i>	A media keyword is included in the package name when the software image cannot be installed using the <code>request system software add</code> command. Values for the media keyword include the following: <ul style="list-style-type: none"> • usb for images you install from a USB drive • net for images you install from the loader prompt
<i>platform</i>	(Optional) Name of the product series, such as mx or ptx.

Table 2: Descriptions of Junos OS Package Name Fields (*Continued*)

Field Name	Description
<i>product</i>	(Optional) Model number or product variant, such as 5e for the QFX Series switches.
<i>architecture</i>	(Optional) CPU architecture of the platform. For example, x86 for Intel CPUs or arm for Advanced RISC Machines CPUs.
<i>application-binary-interface</i>	(Included when <i>architecture</i> is part of the name.) Indicates the “word length” of the CPU architecture. Values include 32 for 32-bit architectures and 64 for 64-bit architectures.
<i>release</i>	Release number. The format of the release number is explained later in this chapter.
<i>edition</i>	Edition of the software package. Software editions are explained later in this chapter.

The software is delivered in signed packages that contain digital signatures, Secure Hash Algorithm (SHA-1), and Message Digest 5 (MD5) checksums. A package is installed only if the checksum within it matches the hash recorded in its corresponding file. Which checksum is used depends on the software version:

- Digital signatures are used when you upgrade or downgrade between Junos OS Release 7.0 and a later version.
- The SHA-1 checksum is used when you upgrade or downgrade between Junos OS Release 6.4 and a later version.
- The MD5 checksum is used when you upgrade or downgrade between Junos OS Release 6.3 or earlier and a later version.

Starting in 2015, the word **signed** appears less frequently after the edition in the filename. But you might still see it in software installation packages. Whether **signed** appears or not, all Junos OS images from Junos OS Release 15.1 on are signed for validation.

Extensions are **tgz**, **gz**, **img**, **iso**, etc.

Junos OS Installation Packages Prefixes

The first part of the installation package filename is a combination of a standard prefix and product designation. Table 3 lists a variety of Junos OS package name prefixes.

Certain hardware platforms run a Junos OS based on an upgraded FreeBSD kernel, greater than FreeBSD 10.x (hereafter called Junos OS with upgraded FreeBSD). To determine if your system uses Junos OS with upgraded FreeBSD, see: [Junos kernel upgrade to FreeBSD 10+](#). Table 3 also indicates the prefixes used for the different platforms running Junos OS with upgraded FreeBSD. For more information about upgrading or downgrading to Junos OS with upgraded FreeBSD, see "[Upgrade and Downgrade for Junos OS with Upgraded FreeBSD](#)" on page 87.

Except where indicated in the table, you install these packages using the `request system software add` CLI command.

Table 3: Installation Package Prefixes

Installation Package Prefix	Description
jinstall*	Junos OS for M Series, MX Series, T Series, TX Matrix, and TX Matrix Plus routers.
jinstall64*	64-bit Junos OS for the JCS1200 Route Reflector, TX Matrix Plus routers with 3D SIBs, and PTX Series Packet Transport Routers.
jinstall-ex*	Junos OS for the EX Series Ethernet Switch portfolio.
jinstall-host-acx5k*	Junos OS with upgraded FreeBSD for the ACX5000 Series routers, which are Linux based; this prefix indicates the image includes the host as well as Junos OS. For example, jinstall-host-acx5k-17.2R1.13-signed.tgz .
jinstall-host-ex*	Junos OS with upgraded FreeBSD for EX4600, which is Linux based; this prefix indicates the image includes the host as well as Junos OS. For example, jinstall-host-ex-4600-17.2R1.13-signed.tgz .
jinstall-host-nfx-2*	<p>Junos OS with upgraded FreeBSD for NFX2xx platforms that are Linux based; this prefix indicates the image includes the host software and Junos OS.</p> <p>For example, jinstall-host-nfx-2-flex-x86-32-17.2R1.13-secure-signed.tgz. See Junos OS Releases Supported on NFX Series Hardware for a list of which platforms use the nfx-2 package.</p>

Table 3: Installation Package Prefixes *(Continued)*

Installation Package Prefix	Description
jinstall-host-nfx-3*	<p>Junos OS with upgraded FreeBSD for NFX platforms that are Linux based; this prefix indicates the image includes the host software and Junos OS.</p> <p>For example, jinstall-host-nfx-3-x86-64-22.4R1.10-secure-signed.tgz. See Junos OS Releases Supported on NFX Series Hardware for a list of which platforms use the nfx-3 package.</p>
jinstall-host-ocx*	Junos OS with upgraded FreeBSD for OCX platforms that are Linux based; this prefix indicates the image includes the host software as well as Junos OS.
jinstall-host-ptx*	Junos OS with upgraded FreeBSD for PTX platforms that are Linux based; this prefix indicates the image includes the host software as well as Junos OS.
jinstall-host-qfx*	<p>Junos OS with upgraded FreeBSD for QFX platforms that are Linux based; this prefix indicates the image includes the host software as well as Junos OS. For example, jinstall-host-qfx-5e-x86-64-17.2R1.13.tgz is a package name for Junos OS on the QFX5100. Starting in Junos OS Release 24.2R1, the package prefix for all models of the QFX5110 is jinstall-host-qfx-5x*. For releases before Release 24.2R1, the package prefix for all models of the QFX5110 is jinstall-host-qfx-5e*.</p>
jinstall-ocx-flex*	OCX Series switches.
jinstall-ppc*	Junos OS for the ACX Series, MX5, MX10, MX40, MX80, and MX104 routers.
junos-arm*	Junos OS with Upgraded FreeBSD for EX2300 and EX3400 switches. For example, junos-arm-32-15.1X53-D50.2.tgz .

Table 3: Installation Package Prefixes *(Continued)*

Installation Package Prefix	Description
junos-arm-media-<i>media-keyword</i>*	<p>Junos OS with Upgraded FreeBSD for EX2300 and EX3400 switches. You install these images using a method other than the request system software add command at the CLI prompt, such as installing from a USB drive or a loader prompt. The media keyword can be one of the following:</p> <ul style="list-style-type: none"> • usb for images you install from a USB drive • net for images you install from the loader prompt <p>For example, junos-install-media-usb-arm-32-15.1X53-D50.2.img or junos-install-media-net-arm-32-15.1X53-D50.2.tgz.</p>
junos-install*	<p>Junos OS with upgraded FreeBSD for EX4100, EX9200, MX Series routers, and SRX Series Firewalls that support Junos OS with upgraded FreeBSD. For example, junos-install-ex-arm-64-22.2R1.3.tgz for EX4100, junos-install-ex92xx-x86-64-17.2R1.13.tgz for EX9200, junos-install-mx-x86-32-15.1R1.9.tgz for MX Series routers, junos-install-srxsme-mips-64-24.4R1.1.tgz for SRX300, SRX320, SRX340, SRX345, and SRX380, and junos-install-srx5000-x86-64-17.3R1.9.tgz for SRX5400, SRX5600, or SRX5800.</p>
junos-install-media-<i>media-keyword</i>*	<p>Junos OS with upgraded FreeBSD for EX4100, EX9200, MX Series routers, and SRX Series Firewalls that support Junos OS with upgraded FreeBSD. You install these images using a method other than the request system software add command at the CLI prompt, such as installing from a USB drive or a loader prompt. The media keyword can be one of the following:</p> <ul style="list-style-type: none"> • usb for images you install from a USB drive • net for images you install from the loader prompt • pxe for images you install using the Preboot Execution Environment (PXE) on the SRX1500, SRX4600, SRX5400, SRX5600, and SRX5800 <p>For example, junos-install-media-usb-mx-x86-32-15.1R1.9.tgz for an MX Series router, junos-install-media-usb-ex-arm-64-22.2R1.3.tgz for EX4100, junos-install-media-usb-ex92xx-17.2R1.13.img.gz for EX9200, junos-install-media-usb-srxsme-mips-64-24.4R1.1.img.gz for SRX300, SRX320, SRX340, SRX345, and SRX380, and junos-install-media-usb-srx5000-x86-64-17.3R1.9.img.gz for SRX5400, SRX5600, and SRX5800.</p>

Table 3: Installation Package Prefixes *(Continued)*

Installation Package Prefix	Description
<code>junos-srx1k3k*</code>	Junos OS for SRX1400, SRX3400 and SRX3600.
<code>junos-srx5000*</code>	Junos OS for SRX5400, SRX5600, and SRX5800.
<code>junos-srxentedge*</code>	Junos OS for SRX1500.
<code>junos-srxhe-x86*</code>	Junos OS for SRX4600.
<code>junos-srxmr*</code>	Junos OS for SRX4100 and SRX4200.
<code>junos-srxsme*</code>	Junos OS for SRX300, SRX320, SRX340, SRX345, SRX380, and SRX550M.
<code>junos-vmhost-install*</code>	Junos OS with upgraded FreeBSD for devices that use VM Host. You use the request <code>vmhost software add</code> CLI command to install these images. For more information about VM Host installation, see "Installing, Upgrading, Backing Up, and Recovery of VM Host" on page 385 .
<code>junos-vmhost-install-media-media-keyword*</code>	<p>Junos OS with upgraded FreeBSD for devices that use VM Host. You install these images using the Preboot Execution Environment (PXE) boot server or the USB drive, and not the request <code>vmhost software add</code> CLI command. The media keyword can be one of the following:</p> <ul style="list-style-type: none"> • <code>usb</code> for images you install from a USB drive • <code>net</code> for images you install from the Preboot Execution Environment (PXE) boot server <p>For more information about this installation method, see "Copying VM Host Installation Package to the PXE Boot Server" on page 392 or "Creating an Emergency Boot Device for Routing Engines with VM Host Support" on page 416.</p>

SEE ALSO

| `show version (Junos OS)`

Junos OS Release Numbers

The release number represents a particular revision of the software that runs on a Juniper Networks routing platform, for example, Junos OS Release 14.1, 14.2, 15.1, or 17.1. Each release has certain new features that complement the software processes that support Internet routing protocols, control the device's interfaces and the device chassis itself, and allow device system management. On the Juniper Networks Support web page, you download software for a particular release number.

In this example, we dissect the format of the software release number to show what it indicates. The generalized format is as follows:

Given the format of

1. *m.nZb.s*

The software release number 17.2R1.13, for example, maps to this format as follows:

- *m* is the main release number of the product, for example, 17.
- *n* is the minor release number of the product, for example, 2.
- *Z* is the type of software release, for example, R for FRS or maintenance release.

For types of software releases, see Table 4.

- *b* is the build number of the product, for example, 1, indicating the FRS rather than a maintenance release..
- *s* is the spin number of the product, for example, 13.

Table 4: Software Release Types

Release Type	Description
R	First revenue ship (FRS) or maintenance release software. R1 is FRS. R2 onward are maintenance releases.
F	Feature velocity release. Feature velocity releases are only in Junos OS Release 15.1.
B	Beta release software.
I	Internal release software. These are private software releases for verifying fixes.

Table 4: Software Release Types (*Continued*)

Release Type	Description
S	Service release software, which are released to customers to solve a specific problem—this release will be maintained along with the life span of the underlying release. The service release number is added after the R number, for example, 14.2R3-S4.4. Here S4 represents the 4th service release on top of 14.2R3 and is the 4th re-spin.
X	<p>Special (eXception) release software. X releases follow a numbering system that differs from the standard release numbering.</p> <p>Starting with Junos OS Release 12.1X44-D10, SRX Series Firewalls follow a special naming convention for Junos OS releases. For more information, refer to the Knowledge Base article KB30092 at https://kb.juniper.net/InfoCenter/index?page=home.</p>



NOTE: Prior to Junos OS Release 11.4, the software release number format for service releases was same as other releases. For example, 10.4S4.2 represented the 4th service release and 2nd re-spin of 10.4.

Junos OS Editions

Editions show up in the installation package name after the release number string and before *signed*.

In releases earlier than Junos OS Release 15.1, installation packages came in several major software package categories or editions, such as domestic, worldwide, or Federal Information Processing Standard (FIPS). For those still using packages with names including these terms, here is what they indicate:

- **domestic**—Junos OS for customers in the United States and Canada and for all other customers with a valid encryption agreement. This edition includes high-encryption capabilities such as IPsec and SSH for data leaving the router or switch. Later images use a null, or empty, edition field for this category.
- **limited**—Junos OS for all other customers. This edition does not include any high-encryption capabilities for data leaving the router or switch. Sometimes referred to as the *Export* edition, starting in Junos OS Release 15.1R1, this category is renamed to the limited edition.
- **fips**—Junos OS that provides advanced network security for customers who need software tools to configure a network of Juniper Networks routers and switches in a Federal Information Processing Standards (FIPS) 140-2 environment. For more information about Junos-FIPS, see FIPS 140-2

Security Compliance. In later images, FIPS, instead of being a separate edition, is an option you select on installation.

Starting with Junos OS 15.1, a simplified edition scheme was started:

- Junos OS with a null (empty) edition field is the standard image for Junos OS.
- **limited**—Version has no cryptographic support and is intended for countries in the Eurasian Customs Union (EACU). These countries have import restrictions on software containing data-plane encryption.

Boot Sequence on Devices with Routing Engines (Junos OS)

IN THIS SECTION

- [Boot Order for Devices | 24](#)
- [Booting from an Alternate Boot Device | 26](#)

Juniper Networks devices start using the installed Junos OS. Bootable copies of Junos OS are stored in various locations: the internal flash disk, the hard drive, the removable media. The following subsections discuss the order of locations checked for a valid bootable operating system.

Boot Order for Devices

Information about the boot order for the various devices with Routing Engines is given in this section in alphabetical order of the device families.



NOTE: For information about which Routing Engines are supported by each device, see https://www.juniper.net/documentation/en_US/release-independent/junos/topics/reference/general/routing-engine-m-mx-t-series-support-by-chassis.html.

The ACX Series routers attempt to boot from the storage media in the following order:

1. USB storage media device
2. Dual, internal NAND flash device (first da0s1, then da0s2)

The router attempts to boot from the storage media in the following order:

MX80 routers attempt to boot from the storage media in the following order:

1. USB media emergency boot device

2. Dual, internal NAND flash device (first da0, then da1)

MX104 routers attempt to boot from the storage media in the following order:

1. USB storage media device
2. Internal NAND flash device (**da0**)

The M Series and MX Series with a Routing Engine that has a solid-state drive (SSD) attempt to boot from the storage media in the following order:

1. USB media emergency boot device (if present)
2. CompactFlash card
3. Solid-state drive (SSD) in the SSD slot 1 or SSD slot 2 (if present)

The M Series and MX Series (except for the MX80 routers and the MX104 routers) routers with a Routing Engine that has a hard disk attempt to boot from the storage media in the following order:

1. Removable media emergency boot device, such as a PC Card (if present)
2. CompactFlash card (if present)
3. Hard disk

The PTX Series Packet Transport Routers attempt to boot from the storage media in the following order:

1. USB media emergency boot device
2. CompactFlash card
3. Solid-state drive (SSD) in the Disk 1 slot (if present)
4. Storage media available on the LAN

The T Series and TX Matrix routers with a Routing Engine that has a hard disk attempt to boot from the storage media in the following order:

1. Removable media emergency boot device, such as a PC Card (if present)
2. CompactFlash card (if present)
3. Hard disk

The T Series routers with a Routing Engine that has a solid-state drive (SSD), and TX Matrix Plus routers attempt to boot from the storage media in the following order:

1. USB media emergency boot device
2. CompactFlash card (if present)

3. Solid-state drive (SSD) in the Disk 1 slot (if present)



NOTE: The Disk 2 slot is not currently supported.

4. Storage media available on the LAN

Booting from an Alternate Boot Device



NOTE: Do not insert an emergency boot device during normal operations. The router does not operate normally when it is booted from an emergency boot device.

If the router boots from an alternate boot device, Junos OS displays a message indicating this when you log in to the router. For example, the following message shows that the software booted from the hard disk (**/dev/ad1s1a**):

```
login: username
Password: password
Last login: date on terminal

--- Junos 8.0 R1 built date
---
--- NOTICE: System is running on alternate media device (/dev/ad2s1a).
```

This situation results when the router detects a problem with the primary boot device—usually the CompactFlash card—that prevents it from booting, and consequently boots from the alternate boot device (the hard disk drive). When this happens, the primary boot device is removed from the list of candidate boot devices. The problem is usually a serious hardware error. We recommend you contact the Juniper Networks Technical Assistance Center (JTAC).



NOTE: On MX104 routers, if the router boots from an alternate boot device, Junos OS does not display any message indicating this when you log in to the router.

When the router boots from the alternate boot device, the software and configuration are only as current as the most recent `request system snapshot` command. However, if the `mirror-flash-on-disk` command was enabled, then the hard disk drive contains a synchronized, mirror image of the compact flash drive and therefore the current software and configuration.

RELATED DOCUMENTATION

[Upgrading and Downgrading to Junos with Upgraded FreeBSD](#)

[Storage Media and Routing Engines \(Junos OS\) | 27](#)

[Junos Software Versions - Suggested Releases to Consider and Evaluate](#)

Storage Media and Routing Engines (Junos OS)

IN THIS SECTION

- [Routing Engines and Storage Media \(Junos OS\) | 27](#)
- [Repartitioning Routing Engine System Storage to Increase the Swap Partition \(Junos OS\) | 28](#)
- [System Memory and Storage Media on Routers \(Junos OS\) | 29](#)
- [Routing Engines and Storage Media Names \(ACX Series, M Series, MX Series, PTX Series, T Series, TX Matrix, TX Matrix Plus, and JCS 1200 Routers\) | 32](#)
- [System Memory and Storage Media for SRX Series Firewalls | 35](#)
- [Accessing USB Storage on PTX1000 Routers | 44](#)

The Routing Engine and the Packet Forwarding Engine (PFE) are the two primary components of Juniper Networks platforms. Junos OS software is installed on the routing engine and it is stored in storage media.

Routing Engines and Storage Media (Junos OS)

Juniper Networks routing platforms are made up of two basic routing components:

- **Routing Engine**—The Routing Engine controls the routing updates and system management.
- **Packet Forwarding Engine (PFE)**—The Packet Forwarding Engine performs Layer 2 and Layer 3 packet switching, route lookups, and packet forwarding.

From a system administration perspective, you install the software onto the Routing Engine and during the installation, the appropriate software is forwarded to other components as necessary. Most Routing Engines include a CompactFlash card that stores Junos OS. On M Series Multiservice Edge Routers; MX240, MX480, and MX960 Universal Routing Platforms; T Series Core Routers; and TX Matrix routers, the system also includes a hard disk or solid-state drive (SSD) that acts as a backup boot drive. PTX Series Packet Transport Routers and the TX Matrix Plus router include a solid-state drive as a backup boot drive.



NOTE: The MX80 router is a single-board router with a built-in Routing Engine and single Packet Forwarding Engine. On an MX80 router, Junos OS is stored on dual, internal NAND flash devices. These devices provide the same functionality as a CompactFlash card and hard disk or solid-state drive (SSD).



NOTE: The ACX Series router is a single board router with a built-in Routing Engine and one Packet Forwarding Engine. The ACX router supports dual-root partitioning, which means that the primary and backup Junos OS images are kept in two independently bootable root partitions. If the primary partition becomes corrupted, the system remains fully functional by booting from the backup Junos OS image located in the other root partition.

On routing platforms with dual Routing Engines, each Routing Engine is independent with regard to upgrading the software. To install new software on both Routing Engines, you need to install the new software on each Routing Engine. On platforms with dual Routing Engines configured for high availability, you can use the unified in-service software upgrade procedure to upgrade the software. For more information about this procedure, see the [High Availability User Guide for Routing Devices](#).

Repartitioning Routing Engine System Storage to Increase the Swap Partition (Junos OS)

You can increase the size of the swap partition by repartitioning the drive (hard disk or solid-state drive [SSD]) on the Routing Engine. This feature is first available in Junos OS Release 10.4R5, 11.1R3, and 11.2R1; in earlier Junos OS releases, the swap partition is not increased by the methods described here.

This behavior applies only to Routing Engines with more than 2 GB of RAM. The new size of the swap partition depends on the size of the drive and the amount of Routing Engine RAM.

- When the drive is 32 GB or less, the swap partition is limited to 8 GB.
- When the drive is larger than 32 GB, the swap partition matches the size of the Routing Engine RAM.

To repartition the drive, perform one of the following actions:

- During the installation of a Junos OS software package (**jinstall***), issue the `request system reboot media disk` command to boot from the drive instead of issuing the `request system reboot` command. The drive is automatically repartitioned. The `request system reboot media disk` command repartitions the drive only during a software upgrade.
- Manually partition the drive by issuing the `request system partition hard-disk` command, and then reboot the router when the command completes.



CAUTION: Repartitioning the drive re-creates the **/config** and **/var** directories in the router file system. Although the contents of **/config** and **/var/db** are preserved, the remaining contents of **/var** are lost. For this reason, we recommend that you back up the **/var** directory before you repartition the SSD on a router with this configuration.

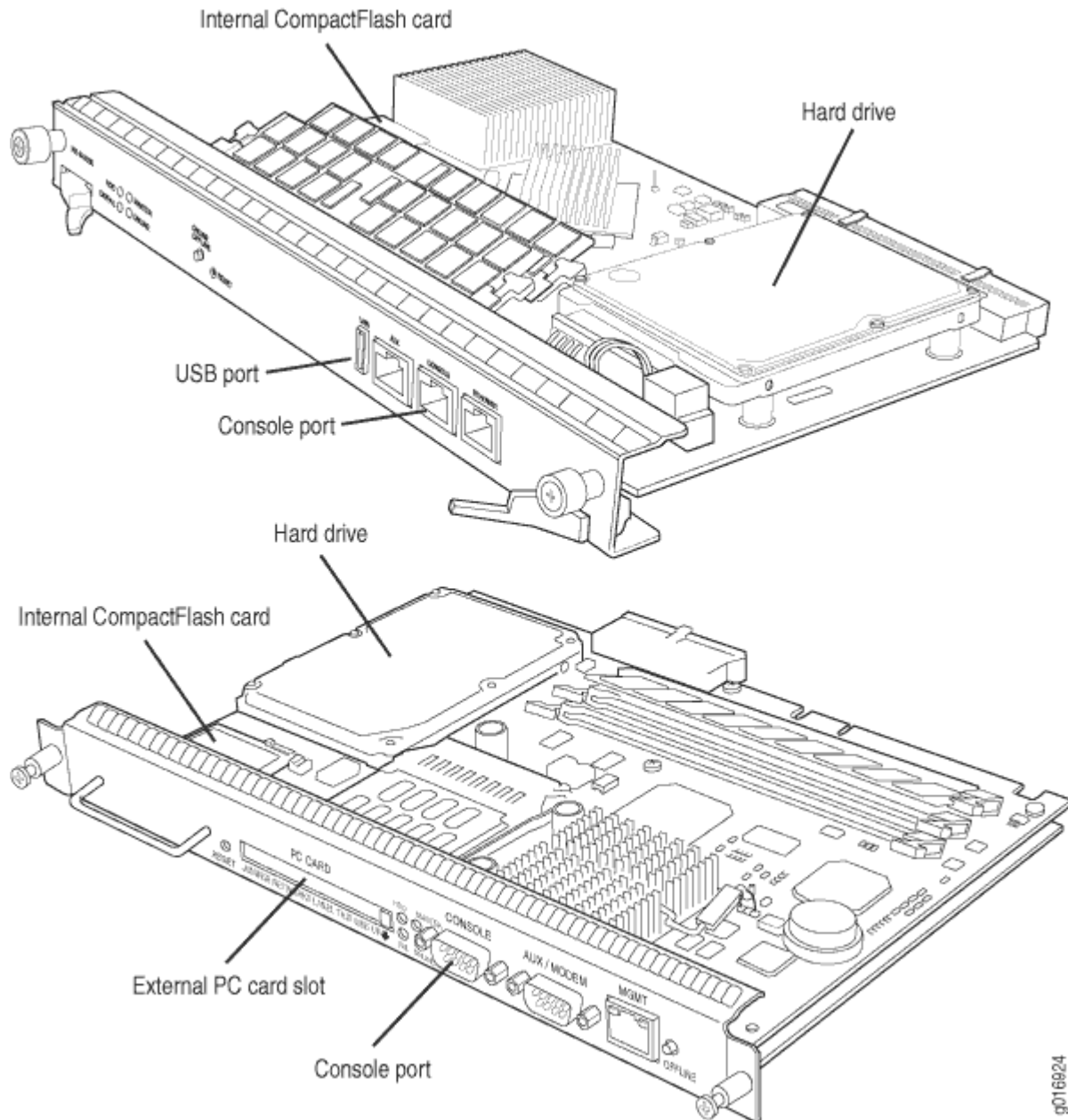
System Memory and Storage Media on Routers (Junos OS)

IN THIS SECTION

- [System Memory | 30](#)
- [Storage Media | 31](#)

[Figure 3 on page 30](#) shows examples of Routing Engines.

Figure 3: Routing Engines



System Memory

Starting with Junos OS Release 9.0, all routing platforms require a minimum of 512 MB of system memory on each Routing Engine. All M7i and M10i routers delivered before December 7, 2007, had 256 MB of memory. These routers require a system memory upgrade before you install Junos OS Release 9.0 or a later release. To determine the amount of memory currently installed on your system, use the `show chassis routing-engine` command in the command-line interface (CLI).

For more information about upgrading your M7i or M10i router, see the Customer Support Center JTAC Technical Bulletin PSN-2007-10-001: <https://www.juniper.net/alerts/viewalert.jsp?txtAlertNumber=PSN-2007-10-001&actionBtn=Search>.

ACX2000 routers are shipped with 2 GB of memory and ACX1000 routers with 1 GB of memory.

Storage Media

Except for the ACX Series, MX80 routers, and MX104 routers, the M Series, MX Series, PTX Series, T Series, TX Matrix, and TX Matrix Plus routers use the following media storage devices:

- CompactFlash card—The CompactFlash card is typically the primary storage device for most routers.



NOTE: M7i and M10i routers using RE-400 are not delivered from the factory with the CompactFlash card installed. In this case, the hard disk is the primary and only boot device. The M7i and M10i routers with RE-400 can be upgraded to include the CompactFlash card.

- Hard disk or solid -state drive—For most routers, a hard disk or solid-state drive is the secondary boot device. When the CompactFlash card is not installed on the router, the hard disk or the solid-state drive becomes the primary boot device. The hard disk or solid-state drive is also used to store system log files and diagnostic dump files.
- Emergency boot device—Depending on the router, the emergency boot device can be a PC card, a USB storage device, or an LS-120 floppy disk.

On MX80 routers, the internal NAND flash devices (first *da0*, then *da1*) act as the primary and secondary boot devices.

On ACX Series routers, the internal NAND flash devices (first *da0s1*, then *da0s2*) act as the primary and secondary boot devices.

Emergency boot devices can be used to revive a routing platform that has a damaged Junos OS. When an emergency boot device is attached to the router, the router attempts to boot from that device before it boots from the CompactFlash card, solid-state drive (SSD), or hard disk.

On an ACX Series router, the emergency boot device is a USB storage device.

On MX104 routers, the internal NAND flash device (*da0*) mounted on the internal eUSB card acts as the primary boot and storage device. On MX104 routers, the emergency boot device is a USB storage device that is plugged into one of the USB ports in the front plate.

When booting from an emergency boot device, the router requests a boot acknowledgment on the console interface. If you enter yes, the emergency boot device repartitions the primary boot device and reloads Junos OS onto the primary boot device. After the loading is complete, the routing platform

requests that you remove the emergency boot device and reboot the system. After the reboot is complete, you must perform an initial configuration of the router before it can be used on your network.



NOTE: For routers with RE-MX-X6, RE-MX-X8, and RE-PTX-X8 Routing Engines, a set of two 64-GB SSDs are available for storage and redundancy. For more information see Storage Partitioning and Redundancy topic in ["Salient Features of the Routing Engines with VM Host Support" on page 372](#) section.

Routing Engines and Storage Media Names (ACX Series, M Series, MX Series, PTX Series, T Series, TX Matrix, TX Matrix Plus, and JCS 1200 Routers)

Table 5 on page 32 specifies the storage media names by Routing Engine. The storage media device names are displayed when the router boots.

Table 5: Routing Engines and Storage Media Names (ACX Series, M Series, MX Series, T Series, TX Matrix, TX Matrix Plus, and JCS 1200 Routers)

Routing Engine	Type of Junos OS	CompactFlash Card	Hard Disk	Solid-State Drive	Removable Media Emergency Boot Device
RE-400-768 (RE5)	FreeBSD 6.x	ad0	ad1	No	ad3
RE-600-2048 (RE3)	FreeBSD 6.x	ad0	ad1	No	ad3
RE-850-1536 (RE-850)	FreeBSD 6.x	ad0	ad1	No	ad3
RE-A-1000-2048 (RE-A-1000)	FreeBSD 6.x	ad0	ad2	No	da0
RE-A-1800x2 (RE-A-1800)	FreeBSD 6.x	ad0	No	Yes SSD1: ad1 SSD2: ad2	da0
RE-S-1300-2048 (RE-S-1300)	FreeBSD 6.x	ad0	ad2	No	da0

Table 5: Routing Engines and Storage Media Names (ACX Series, M Series, MX Series, T Series, TX Matrix, TX Matrix Plus, and JCS 1200 Routers) (Continued)

Routing Engine	Type of Junos OS	CompactFlash Card	Hard Disk	Solid-State Drive	Removable Media Emergency Boot Device
RE-S-1800x2 RE-S-1800x4 (RE-S-1800)	FreeBSD 6.x	ad0	No	Yes SSD1: ad1 SSD2: ad2	da0
	FreeBSD 10.x/ 11.x				
RE-B-1800X1-4G-S	FreeBSD 6.x	ad0	No	Yes SSD1: ad1	da0
RE-1600-2048 (RE4)	FreeBSD 6.x	ad0	ad1	No	ad3 and ad4
RE-A-2000-4096 (RE-A-2000)	FreeBSD 6.x	ad0	ad2	No	da0
RE-S-2000-4096 (RE-S-2000)	FreeBSD 6.x	ad0	ad2	No	da0
RE-MX-104	FreeBSD 6.x	No	da0	No	da1 and da2
RE-DUO-C2600-16G (RE-DUO-2600)	FreeBSD 6.x	ad0	No	ad1	da0
RE-DUO-C1800-8G- (RE-DUO-1800)	FreeBSD 6.x	ad0	No	ad1	da0
RE-DUO-C1800-16G	FreeBSD 6.x	ad0	No	ad1	da0

Table 5: Routing Engines and Storage Media Names (ACX Series, M Series, MX Series, T Series, TX Matrix, TX Matrix Plus, and JCS 1200 Routers) (Continued)

Routing Engine	Type of Junos OS	CompactFlash Card	Hard Disk	Solid-State Drive	Removable Media Emergency Boot Device
RE-JCS1200-1x2330	FreeBSD 6.x	da0	da1	No	da2
RE-PTX-X8-64G	FreeBSD 6.x	No	No	Yes SSD1: sda SSD2: sdb	da0
RE-S-X6-64G	FreeBSD 6.x	No	No	Yes SSD1: sda SSD2: sdb	da0
REMX2K-X8-64G	FreeBSD 6.x	No	No	Yes SSD1: sda SSD2: sdb	da0



NOTE: On MX80 routers, the Routing Engine is a built-in device and has no model number. The dual internal NAND flash devices are da0 and da1. The USB storage device is da2.



NOTE: On ACX Series routers, the Routing Engine is a built-in device which does not have a model number. The dual internal NAND flash devices are da0s1 and da0s2. The USB storage device is da0s2a. Use the `show chassis hardware models` command to obtain the field-replaceable unit (FRU) model number—for example, ACX2000BASE-DC for the ACX2000 router.

To view the storage media currently available on your system, use the CLI `show system storage` command.

SEE ALSO

Supported Routing Engines by Router
Routing Engine Specifications
RE-S-1300 Routing Engine Description
RE-S-2000 Routing Engine Description
RE-S-1800 Routing Engine Description
JCS1200 Routing Engine Description

System Memory and Storage Media for SRX Series Firewalls

IN THIS SECTION

- [SRX Series Firewalls Overview | 35](#)
- [System Memory | 43](#)
- [Storage Media | 43](#)

SRX Series Firewalls Overview

Figure 4 on page 35 shows the front panel of an SRX345 Firewall.

Figure 4: SRX345 Firewall Front Panel

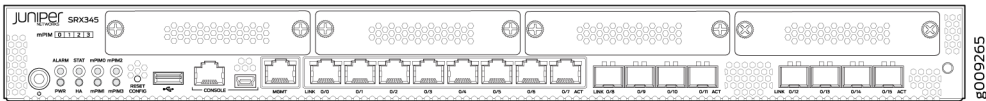


Figure 5 on page 35 shows an example of an SRX1500 Firewall.

Figure 5: SRX1500 Firewall Front Panel

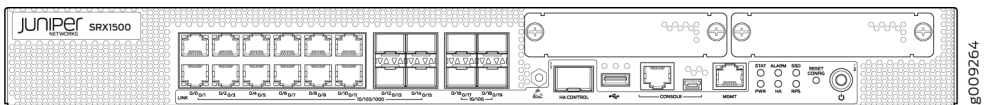
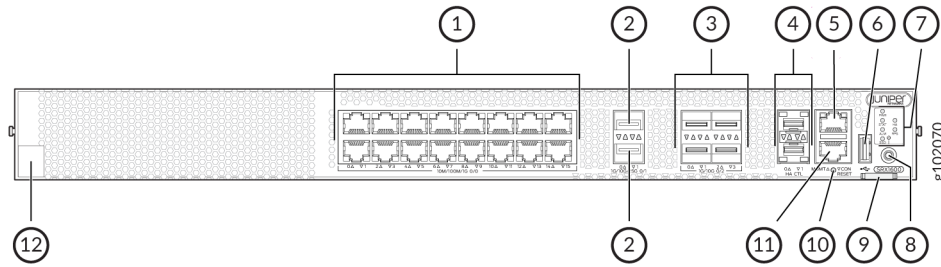


Figure 6: SRX1600 Firewall Front Panel



The below table lists and describes the front panel components of the firewall.

Table 6: Components on the Front Panel of the SRX1600

Callout	Component (Label on the Chassis)	Description
1	Ethernet ports	Sixteen 10/100/1000 BASE-T ports (PIC 0 ports)
2	SFP28 ports	Two 1/10/25 GbE SFP28 MACsec ports for network traffic (PIC 1 ports)
3	SFP+ ports	Four 1/10 GbE SFP+ MACsec ports for network traffic (PIC 2 ports)
4	Chassis cluster ports (HA)	Two 1 GbE SFP chassis cluster control CTL ports with MACsec support
5	Management port (MGMT)	1 GbE RJ-45 port
6	USB port	One USB 3.0 Type A port that accepts a USB storage device
7	Chassis LEDs	Indicate component and system status and troubleshooting information at a glance.
8	Power button	Power button

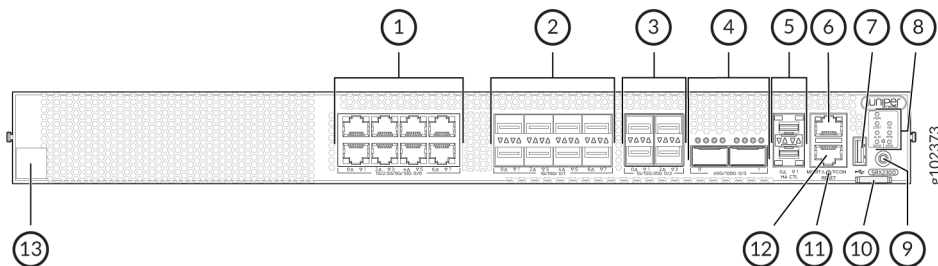
Table 6: Components on the Front Panel of the SRX1600 (Continued)

Callout	Component (Label on the Chassis)	Description
9	Pull tab	Contains the CLEI code and serial number of the device
10	RESET	Reset button. To reset the system, press and hold the RESET button for around 250 ms.
11	Console port (CON)	You can connect a laptop to the SRX1600 to manage the CLI. The port uses an RJ-45 serial connection and supports the RS-232 (EIA-232) standard.
12	Claim code	You can use the QR code to claim and onboard your device to Juniper Security Director.



NOTE: The BASE-T PIC 0 ports (ge-0/0/0 to ge-0/0/15) support autonegotiation. The SFP28 PIC 1 ports (et-0/1/0 to et-0/1/1) do not support SFP-T or autonegotiation. The SFP+ PIC 2 ports (xe-0/2/0 to xe-0/2/3) support 1 GbE SFP-T but do not support autonegotiation.

Figure 7: SRX2300, SRX4120 Firewall Front Panel



The below table lists and describes the front panel components of the firewall.

Table 7: Components on the Front Panel of the SRX2300, SRX4120

Callout	Component (Label on the Chassis)	Description
1	Ethernet ports	Eight 1/2.5/5/10-Gigabit Ethernet MACsec ports for network traffic (PIC 0 ports)
2	SFP+ ports	Eight 1/10-Gigabit Ethernet SFP+ MACsec ports for network traffic (PIC 1 ports)
3	SFP28 ports	Four 1/10/25-Gigabit Ethernet SFP28 MACsec ports for network traffic (PIC 2 ports)
4	QSFP28	Two 40/100-Gigabit Ethernet QSFP28 MACsec ports for network traffic (PIC 3 ports)
5	Chassis cluster ports (HA)	Two 1-Gigabit Ethernet SFP chassis cluster control CTL ports with MACsec support
6	Management port (MGMT)	1-Gigabit Ethernet RJ-45 port
7	USB port	One USB 3.0 Type A port that accepts a USB storage device.
8	Chassis LEDs	Indicate component and system status and troubleshooting information at a glance.
9	Power button	Power button
10	Pull tab	Contains the CLEI code and serial number of the device.
11	RESET	Reset button. To reset the system, press and hold the RESET button for around 250 ms.
12	Console port (CON)	You can connect a laptop to the SRX2300, SRX4120 for CLI management. The port uses an RJ-45 serial connection and supports the RS-232 (EIA-232) standard.

Table 7: Components on the Front Panel of the SRX2300, SRX4120 (Continued)

Callout	Component (Label on the Chassis)	Description
13	Claim code	You can use the QR code to claim and onboard your device to Juniper Security Director.

NOTE: The BASE-T PIC 0 ports (mge-0/0/0 to mge-0/0/7) support autonegotiation. The SFP+ PIC 1 ports (xe-0/1/0 to xe-0/1/7) support 1 GbE SFP-T but do not support autonegotiation. The SFP28 PIC 2 ports (et-0/2/0 to et-0/2/3) and QSFP28 PIC 3 ports (et-0/3/0 to et-0/3/1) do not support SFP-T or autonegotiation.

Figure 8 on page 39 shows an example of an SRX4200 Firewall.

Figure 8: SRX4200 Firewall Front Panel

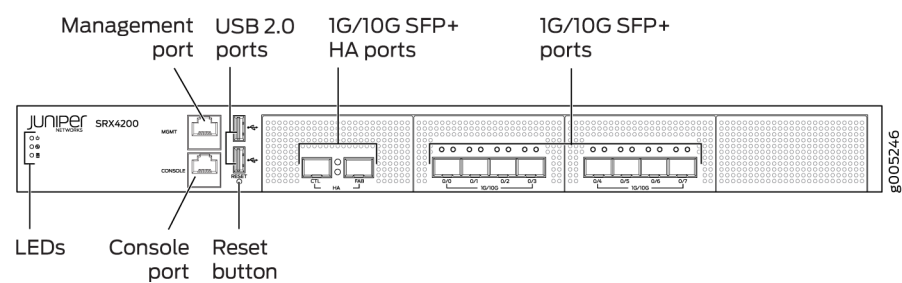
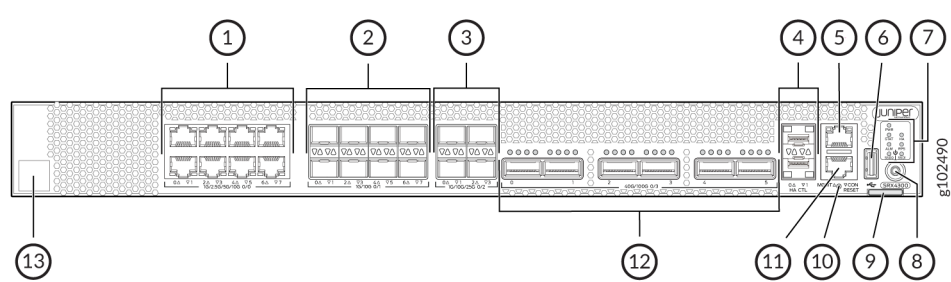


Figure 10 on page 41 shows an example of an SRX4600 Firewall.

Figure 9: SRX4300 Firewall Front Panel



The below table lists and describes the front panel components of the firewall.

Table 8: Components on the Front Panel of the SRX4300

Callout	Component (Label on the Chassis)	Description
1	Ethernet ports	Eight 1/2.5/5/10-Gigabit Ethernet MACsec ports for network traffic.
2	SFP+ ports	Eight 1/10-Gigabit Ethernet SFP+ MACsec ports for network traffic.
3	SFP28 ports	Four 1/10/25-Gigabit Ethernet SFP28 MACsec ports for network traffic.
4	Chassis cluster ports (HA)	Two 1-Gigabit Ethernet SFP chassis cluster control CTL ports with MACsec support
5	Management port (MGMT)	1-Gigabit Ethernet RJ-45 port
6	USB port	One USB 3.0 Type A port that accepts a USB storage device.
7	Chassis LEDs	Indicate component and system status and troubleshooting information at a glance.
8	Power button	Power button
9	Pull tab	Contains the CLEI code and serial number of the device.
10	RESET	Reset button. To reset the system, press and hold the RESET button for around 250 ms.
11	Console port (CON)	You can connect a laptop to the SRX4300 for CLI management. The port uses an RJ-45 serial connection and supports the RS-232 (EIA-232) standard.
12	QSFP28 ports	Six 4x10/4x25/2x50/40/100-Gigabit Ethernet QSFP28 MACsec ports for network traffic.

Table 8: Components on the Front Panel of the SRX4300 (Continued)

Callout	Component (Label on the Chassis)	Description
13	Claim code	You can use the QR code to claim and onboard your device to Juniper Security Director.

NOTE: The BASE-T PIC 0 ports (mge-0/0/0 to mge-0/0/7) support autonegotiation. The SFP+ PIC 1 ports (xe-0/1/0 to xe-0/1/7) support 1 GbE SFP-T but do not support autonegotiation. The SFP28 PIC 2 ports (et-0/2/0 to et-0/2/3) and QSFP28 PIC 3 ports (et-0/3/0 to et-0/3/5) do not support SFP-T or autonegotiation.

Figure 10: SRX4600 Firewall Front Panel

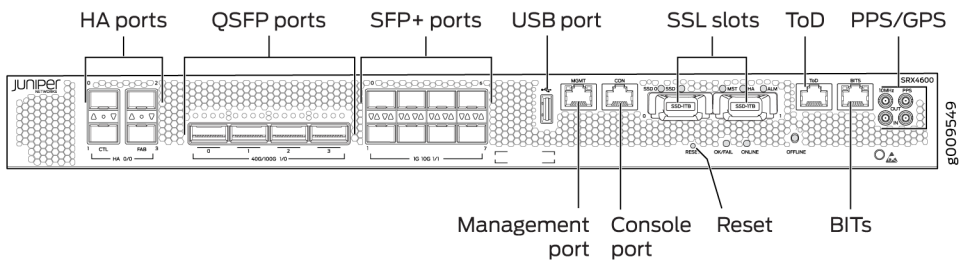
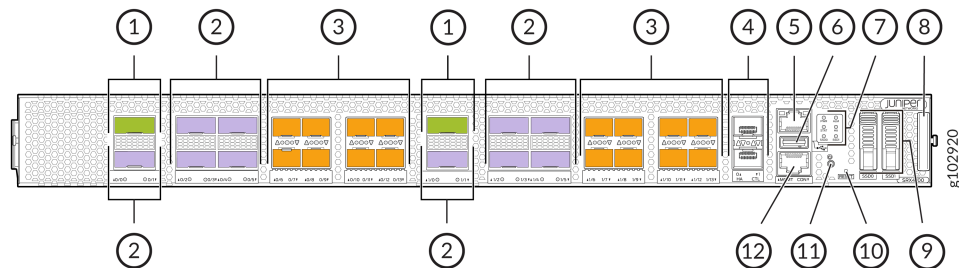


Figure 11: Front Panel Components of SRX4700



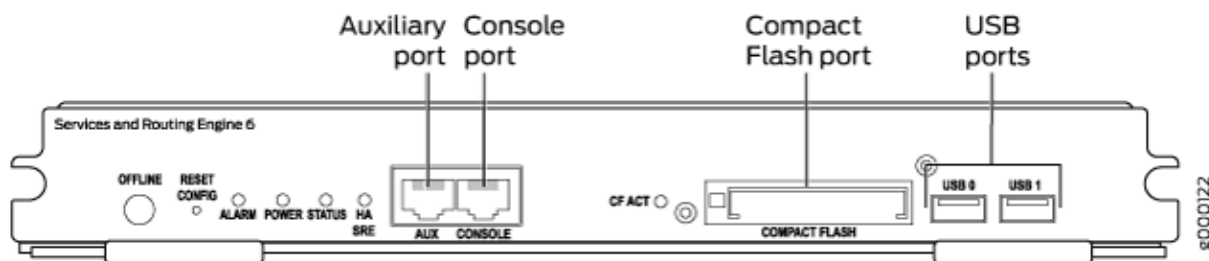
The below table lists and describes the front panel components of the firewall.

Table 9: Components on the Front Panel of the SRX4700

Callout	Component (Label on the Chassis)	Description
1	QSFP56-DD ports	Two 400-Gigabit Ethernet QSFP56-DD MACsec ports for network traffic.
2	QSFP28 ports	Ten 100-Gigabit Ethernet QSFP28 MACsec ports for network traffic.
3	SFP56 ports	Sixteen 50-Gigabit Ethernet SFP56 MACsec ports for network traffic.
4	Chassis cluster ports (HA)	Two 1-Gigabit Ethernet SFP chassis cluster control CTL ports with MACsec support
5	Management port (MGMT)	1-Gigabit Ethernet RJ-45 port
6	USB port	One USB 3.0 Type A port that accepts a USB storage device.
7	Chassis LEDs	Indicate component and system status and troubleshooting information at a glance.
8	Pull-out Information Tab	Contains the serial number.
9	SSD	2x1TB M.2 SSD or 1x1TB M.2 SSD + 1x2TB M.2 SSD
10	RESET	Reset button. To reset the system, press and hold the RESET button for around 250 ms.
11	Power button	Power button
12	Console port (CON)	You can connect a laptop to the SRX4700 for CLI management. The port uses an RJ-45 serial connection and supports the RS-232 (EIA-232) standard.

Figure 12 on page 43 shows an example of an SRX5800 Firewall Routing Engine.

Figure 12: SRX5800 Firewall Routing Engine



System Memory

The amount of free disk space necessary to upgrade a device with a new version of Junos OS can vary from one release to another for different SRX Series Firewalls. Check the Junos OS software version you are installing to determine the free disk space requirements.

To determine the amount of free disk space on the SRX Series Firewall, issue the `show system storage detail` command. The command output displays statistics about the amount of free disk space in the device file systems.

Storage Media

The SRX300, SRX320, SRX340, and SRX345 Firewalls can boot from the following storage media (in the order of priority):

- Internal NAND flash device mounted on the internal eUSB card (default; always present)
- USB storage key (alternate)
- External SSD (SRX340 and SRX345 devices)

The SRX380 Firewall can boot from the following storage media (in the order of priority):

- Internal SSD (default; always present)
- USB storage key (alternate)

SRX1500 Firewalls use the following media storage devices:

- Internal eSATA flash disk (default; always present)
- SSD

SRX1400, SRX3400, SRX3600, SRX5400, SRX5600, and SRX5800 Firewalls use the following media storage devices:

- The CompactFlash card in the Routing Engine

- The hard disk in the Routing Engine



NOTE: You can also use a Junos OS image stored on a USB flash drive that you insert into the Routing Engine faceplate.

The SRX4100 and SRX4200 Firewalls include the following storage media:

- Internal eSATA flash disk (default; always present)
- SSD

The SRX4600 Firewalls include the following storage media:

- Internal eSATA flash disk (default; always present)
- SSD

[Table 10 on page 44](#) specifies the storage media names used by the SRX Series Firewalls. The storage media device names are displayed as the firewall boots.

Table 10: Storage Media Names

Device	Internal CompactFlash Card	USB Storage Media Devices
SRX Series Firewall	da0	da1

To view the storage media currently available on your system, use the CLI `show system storage` command.

SEE ALSO

| [Verifying PIC Combinations \(Junos OS\) | 85](#)

Accessing USB Storage on PTX1000 Routers

On PTX1000 routers, you can only view the USB storage information from Junos OS by using the CLI command `show vmhost hardware`, but cannot access it. However, you can access the USB storage information from the Linux host. From the Linux host, you can also send the USB storage device information with images across different sites where PTX1000 routers are deployed.

To access the USB storage device information on PTX1000 routers:

1. In Junos OS, ensure that the PTX1000 USB image to be copied to the USB storage device is present on the **var/tmp** folder of Junos OS. To copy the image from the **/var/tmp** directory of Junos OS to the **/var/tmp** directory of a Linux host, execute the following command on Junos OS:

```
user@host # vhclient rcp /var/tmp image-name
```

2. On the Linux host shell, execute the following command:

```
user@host # vhclient -s
dd if=/var/tmp/copied-image-name of=/dev/sdc bs=4M
sync
sync
```

In the command above, **/dev/sdc** is the USB storage device detected by the Linux host. You can determine the name of the USB storage device from host logs as shown in the sample below:

```
user@host # dmesg
...
...
[645888.884431] usb 1-1.2: new high-speed USB device number 5 using ehci-pci
[645889.131217] usb-storage 1-1.2:1.0: USB Mass Storage device detected
[645889.131275] scsi8 : usb-storage 1-1.2:1.0
[645890.134290] scsi 8:0:0:0: Direct-Access    JetFlash Transcend 8GB      8.07 PQ: 0 ANSI: 2
[645890.134456] sd 8:0:0:0: Attached scsi generic sg2 type 0

[645890.135908] sd 8:0:0:0: [sdc] 15687680 512-byte logical blocks: (8.03 GB/7.48 GiB)
```

In this example, **sdc** is the name of the USB storage device.



NOTE: The **/var/tmp** directory of a Linux host is mounted on the RAM (at the **ramfs** location), which is volatile storage, and is thus lost when you perform power cycling of or reboot the device. However, the Junos OS **/var/tmp** directory resides on the physical (nonvolatile) hard disk and thus exists even after rebooting or power cycling.

SEE ALSO

[Creating an Emergency Boot Device for Routers](#) | 335

RELATED DOCUMENTATION

| [Installing Software on Routing Devices \(Junos OS\)](#) | 102

Prepare for Installation and Upgrade

IN THIS CHAPTER

- [Preparing for Software Installation and Upgrade \(Junos OS\) | 47](#)
- [Managing Disk Space for Junos OS Installation | 84](#)
- [Verifying PIC Combinations \(Junos OS\) | 85](#)

Preparing for Software Installation and Upgrade (Junos OS)

IN THIS SECTION

- [Upgrade or Reinstall Junos OS | 48](#)
- [Validating the Configuration Image Before Upgrading or Downgrading the Software \(Junos OS\) | 63](#)
- [Ensure Sufficient Disk Space for Junos OS Upgrades on SRX Series Firewalls | 64](#)
- [Verifying Junos OS and Boot Loader Software Versions on an EX Series Switch | 66](#)
- [Access Juniper Support | 72](#)
- [Downloading Software \(Junos OS\) | 73](#)
- [Reinstall Junos OS | 79](#)
- [Reconfigure Junos OS | 80](#)

Before you install or upgrade Junos OS, you must ensure some basic checks such as sufficient disk space availability and backing up configurations in place.

Upgrade or Reinstall Junos OS

IN THIS SECTION

- [Checklist for Reinstalling Junos OS | 48](#)
- [Log the Software Version Information \(Junos OS\) | 50](#)
- [Log the Hardware Version Information \(Junos OS\) | 52](#)
- [Log the Chassis Environment Information \(Junos OS\) | 54](#)
- [Log the System Boot-Message Information \(Junos OS\) | 55](#)
- [Log the Active Configuration \(Junos OS\) | 57](#)
- [Log the Interfaces on the Router \(Junos OS\) | 58](#)
- [Log the BGP, IS-IS, and OSPF Adjacency Information \(Junos OS\) | 60](#)
- [Log the System Storage Information \(Junos OS\) | 62](#)

Checklist for Reinstalling Junos OS

[Table 11 on page 48](#) provides links and commands for reinstalling Junos OS.

Table 11: Checklist for Reinstalling Junos OS

Tasks	Command or Action
Before You Reinstall Junos OS	
"Log the Software Version Information (Junos OS)" on page 50	show version save <i>filename</i>
"Log the Hardware Version Information (Junos OS)" on page 52	show chassis hardware save <i>filename</i>
"Log the Chassis Environment Information (Junos OS)" on page 54	show chassis environment save <i>filename</i>
"Log the System Boot-Message Information (Junos OS)" on page 55	show system boot-messages save <i>filename</i>

Table 11: Checklist for Reinstalling Junos OS (*Continued*)

Tasks	Command or Action
"Log the Active Configuration (Junos OS)" on page 57	show configuration save <i>filename</i>
"Log the Interfaces on the Router (Junos OS)" on page 58	show interface terse save <i>filename</i>
"Log the BGP, IS-IS, and OSPF Adjacency Information (Junos OS)" on page 60	show bgp summary save <i>filename</i> show isis adjacency brief save <i>filename</i> show ospf neighbor brief save <i>filename</i>
"Log the System Storage Information (Junos OS)" on page 62	show system storage save <i>filename</i>
Back Up the Currently Running and Active File System	request system snapshot
"Reinstall Junos OS" on page 79	Insert your removable medium and reboot the system.
"Reconfigure Junos OS" on page 80	
Configure Host Names, Domain Names, and IP Addresses	Log in as root. Start the CLI. Enter configuration mode: configure set system host-name <i>host-name</i> set system domain-name <i>domain-name</i> set interfaces fxp0 unit 0 family inet address <i>address/prefix-length</i> set system backup-router <i>address</i> set system name-server <i>address</i>
Protect Network Security by Configuring the Root Password	set system root-authentication plain-text-password set system root-authentication encrypted-password password set system root-authentication ssh-rsa key commit exit
Check Network Connectivity	ping <i>address</i>

Table 11: Checklist for Reinstalling Junos OS *(Continued)*

Tasks	Command or Action
"Copy Backup Configurations and Restore Saved Configurations" on page 356	<div>file copy var/tmp configure</div> <div>[edit] load merge /config/<i>filename</i> or load replace /config/<i>filename</i></div> <div>[edit] commit</div>
After You Reinstall Junos OS	
Compare Information Logged Before and After the Reinstall	<div>show version save <i>filename</i></div> <div>show chassis hardware save <i>filename</i> show chassis environment save <i>filename</i></div> <div>show system boot-messages save <i>filename</i> show configuration save <i>filename</i></div> <div>show interfaces terse save <i>filename</i></div> <div>show bgp summary show isis adjacency brief show ospf neighbor brief save <i>filename</i></div> <div>show system storage save <i>filename</i></div>
Back Up the New Software	request system snapshot

Log the Software Version Information (Junos OS)

IN THIS SECTION

Purpose | 50

Action | 51

Meaning | 52

Purpose

The purpose of this action is to log the Junos OS version information.

Action

Use the following Junos OS CLI operational mode command:

```
user@host> show version | save filename
```

Sample Output

```
user@host> show version | save test
Wrote 39 lines of output to 'test'

user@host> show version
Hostname: my-router.net
Model: m10
JUNOS Base OS boot [5.0R5]
JUNOS Base OS Software Suite [5.0R5]
JUNOS Kernel Software Suite [5.0R5]
JUNOS Routing Software Suite [5.0R5]
JUNOS Packet Forwarding Engine Support [5.0R5]
JUNOS Crypto Software Suite [5.0R5]
JUNOS Online Documentation [5.0R5]
KERNEL 5.0R5 #0 built by builder on 2002-03-02 05:10:28 UTC
MGD release 5.0R5 built by builder on 2002-03-02 04:45:32 UTC
CLI release 5.0R5 built by builder on 2002-03-02 04:44:22 UTC
CHASSISD release 5.0R5 built by builder on 2002-03-02 04:43:37 UTC
DCD release 5.0R5 built by builder on 2002-03-02 04:42:47 UTC
RPD release 5.0R5 built by builder on 2002-03-02 04:46:17 UTC
SNMPD release 5.0R5 built by builder on 2002-03-02 04:52:26 UTC
MIB2D release 5.0R5 built by builder on 2002-03-02 04:45:37 UTC
APSD release 5.0R5 built by builder on 2002-03-02 04:43:31 UTC
VRRPD release 5.0R5 built by builder on 2002-03-02 04:52:34 UTC
ALARM release 5.0R5 built by builder on 2002-03-02 04:43:24 UTC
PFED release 5.0R5 built by builder on 2002-03-02 04:46:06 UTC
CRAFTD release 5.0R5 built by builder on 2002-03-02 04:44:30 UTC
SAMPLED release 5.0R5 built by builder on 2002-03-02 04:52:20 UTC
ILMID release 5.0R5 built by builder on 2002-03-02 04:45:21 UTC
BPRELAYD release 5.0R5 built by builder on 2002-03-02 04:42:41 UTC
RMOPD release 5.0R5 built by builder on 2002-03-02 04:46:11 UTC
jkernel-dd release 5.0R5 built by builder on 2002-03-02 04:41:07 UTC
jroute-dd release 5.0R5 built by builder on 2002-03-02 04:41:21 UTC
jdocs-dd release 5.0R5 built by builder on 2002-03-02 04:39:11 UTC
```

Meaning

The sample output shows the hostname, router model, and the different Junos OS packages, processes, and documents.

Log the Hardware Version Information (Junos OS)

IN THIS SECTION

- [Purpose | 52](#)
- [Action | 52](#)
- [Sample Output | 53](#)
- [Meaning | 54](#)

Purpose

You should log hardware version information in the rare event that a router cannot successfully reboot and you cannot obtain the Routing Engine serial number. The Routing Engine serial number is necessary for Juniper Networks Technical Assistance Center (JTAC) to issue a return to manufacturing authorization (RMA). Without the Routing Engine serial number, an onsite technician must be dispatched to issue the RMA.

Action

To log the router chassis hardware version information, use the following Junos OS CLI operational mode command:

```
user@host> show chassis hardware | save filename
```

Sample Output

The output for the M-series routers varies depending on the chassis components of each router. All routers have a chassis, midplanes or backplanes, power supplies, and Flexible PIC Concentrators (FPCs). Refer to the hardware guides for information about the different chassis components.

```
user@host> show chassis hardware | save test
```

```
Wrote 43 lines of output to 'test'
```

```
user@host> show chassis hardware
```

Item	Version	Part number	Serial number	Description
Chassis			101	M160
Midplane	REV 02	710-001245	S/N AB4107	
FPM CMB	REV 01	710-001642	S/N AA2911	
FPM Display	REV 01	710-001647	S/N AA2999	
CIP	REV 02	710-001593	S/N AA9563	
PEM 0	Rev 01	740-001243	S/N KJ35769	DC
PEM 1	Rev 01	740-001243	S/N KJ35765	DC
PCG 0	REV 01	710-001568	S/N AA9794	
PCG 1	REV 01	710-001568	S/N AA9804	
Host 1			da000004f8d57001	teknor
MCS 1	REV 03	710-001226	S/N AA9777	
SFM 0 SPP	REV 04	710-001228	S/N AA2975	
SFM 0 SPR	REV 02	710-001224	S/N AA9838	Internet Processor I
SFM 1 SPP	REV 04	710-001228	S/N AA2860	
SFM 1 SPR	REV 01	710-001224	S/N AB0139	Internet Processor I
FPC 0	REV 03	710-001255	S/N AA9806	FPC Type 1
CPU	REV 02	710-001217	S/N AA9590	
PIC 1	REV 05	750-000616	S/N AA1527	1x OC-12 ATM, MM
PIC 2	REV 05	750-000616	S/N AA1535	1x OC-12 ATM, MM
PIC 3	REV 01	750-000616	S/N AA1519	1x OC-12 ATM, MM
FPC 1	REV 02	710-001611	S/N AA9523	FPC Type 2
CPU	REV 02	710-001217	S/N AA9571	
PIC 0	REV 03	750-001900	S/N AA9626	1x STM-16 SDH, SMIR
PIC 1	REV 01	710-002381	S/N AD3633	2x G/E, 1000 BASE-SX
FPC 2				FPC Type OC192
CPU	REV 03	710-001217	S/N AB3329	
PIC 0	REV 01			1x OC-192 SM SR-2

Meaning

The sample output shows the hardware inventory for an M160 router with a chassis serial number of 101. For each component, the output shows the version number, part number, serial number, and description.

Log the Chassis Environment Information (Junos OS)

IN THIS SECTION

- [Action | 54](#)
- [Sample Output | 54](#)
- [Meaning | 55](#)

Action

To log the router chassis environment information, use the following Junos OS CLI operational mode command:

```
user@host> show chassis environment | save filename
```

Sample Output

The following example shows output from the `show chassis environment` command for an M5 router:

```
user@m5-host> show chassis environment | save test
Wrote 14 lines of output to 'test'

user@m5-host> show chassis environment
```

Class	Item	Status	Measurement
Power	Power Supply A	OK	
	Power Supply B	OK	
Temp	FPC Slot 0	OK	32 degrees C / 89 degrees F
	FEB	OK	31 degrees C / 87 degrees F
	PS Intake	OK	26 degrees C / 78 degrees F
	PS Exhaust	OK	31 degrees C / 87 degrees F
Fans	Left Fan 1	OK	Spinning at normal speed

Left Fan 2	OK	Spinning at normal speed
Left Fan 3	OK	Spinning at normal speed
Left Fan 4	OK	Spinning at normal speed

Meaning

The sample output shows the environmental information about the router chassis, including the temperature and information about the fans, power supplies, and Routing Engine.

Log the System Boot-Message Information (Junos OS)

IN THIS SECTION

- [Action | 55](#)
- [Sample Output | 55](#)
- [Meaning | 57](#)

Action

To log the system boot-message information, use the following Junos OS CLI operational mode command:

```
user@host> show system boot-messages | save filename
```

Sample Output

```
user@host> show system boot-messages | save test
Wrote 80 lines of output to 'test'

user@host> show system boot-messages
Copyright (c) 1992-1998 FreeBSD Inc.
Copyright (c) 1996-2000 Juniper Networks, Inc.
All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California. All rights reserved.
```

```

JUNOS 4.1-20000216-Zf8469 #0: 2000-02-16 12:57:28 UTC
  tlim@device1.example.com:/p/build/20000216-0905/4.1/release_kernel/sys/compile/GENERIC
CPU: Pentium Pro (332.55-MHz 686-class CPU)
  Origin = "GenuineIntel"  Id = 0x66a  Stepping=10

Features=0x183f9ff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,SEP,MTRR,PGE,MCA,CMOV,<b16>,<b17>,MMX,<b24>
>
Teknor CPU Card Recognized
real memory  = 805306368 (786432K bytes)
avail memory = 786280448 (767852K bytes)
Probing for devices on PCI bus 0:
chip0 <generic PCI bridge (vendor=8086 device=7192 subclass=0)> rev 3 class 60000 on pci0:0:0
chip1 <Intel 82371AB PCI-ISA bridge> rev 1 class 60100 on pci0:7:0
chip2 <Intel 82371AB IDE interface> rev 1 class 10180 on pci0:7:1
chip3 <Intel 82371AB USB interface> rev 1 class c0300 int d irq 11 on pci0:7:2
smb0 <Intel 82371AB SMB controller> rev 1 class 68000 on pci0:7:3
pcic0 <TI PCI-1131 PCI-CardBus Bridge> rev 1 class 60700 int a irq 15 on pci0:13:0
TI1131 PCI Config Reg: [pci only][FUNC0 pci int]
pcic1 <TI PCI-1131 PCI-CardBus Bridge> rev 1 class 60700 int b irq 12 on pci0:13:1
TI1131 PCI Config Reg: [pci only][FUNC1 pci int]
fxp0 <Intel EtherExpress Pro 10/100B Ethernet> rev 8 class 20000 int a irq 12 on pci0:16:0
chip4 <generic PCI bridge (vendor=1011 device=0022 subclass=4)> rev 4 class 60400 on pci0:17:0
fxp1 <Intel EtherExpress Pro 10/100B Ethernet> rev 8 class 20000 int a irq 10 on pci0:19:0
Probing for devices on PCI bus 1:mcs0 <Miscellaneous Control Subsystem> rev 12 class ff0000 int
a irq 12 on pci1:13:0
fxp2 <Intel EtherExpress Pro 10/100B Ethernet> rev 8 class 20000 int a irq 10 on pci1:14:0
Probing for devices on the ISA bus:
sc0 at 0x60-0x6f irq 1 on motherboard
sc0: EGA color <16 virtual consoles, flags=0x0>
ed0 not found at 0x300
ed1 not found at 0x280
ed2 not found at 0x340
psm0 not found at 0x60
sio0 at 0x3f8-0x3ff irq 4 flags 0x20010 on isa
sio0: type 16550A, console
sio1 at 0x3e8-0x3ef irq 5 flags 0x20000 on isa
sio1: type 16550A
sio2 at 0x2f8-0x2ff irq 3 flags 0x20000 on isa
sio2: type 16550A
pcic0 at 0x3e0-0x3e1 on isa
PC-Card ctlr(0) TI PCI-1131 [CardBus bridge mode] (5 mem & 2 I/O windows)
pcic0: slot 0 controller I/O address 0x3e0
np0 flags 0x1 on motherboard

```



```

npx0: INT 16 interface
fdc0: direction bit not set
fdc0: cmd 3 failed at out byte 1 of 3
fdc0 not found at 0x3f0
wdc0 at 0x1f0-0x1f7 irq 14 on isa
wdc0: unit 0 (wd0): <SunDisk SDCFB-80>, single-sector-i/o
wd0: 76MB (156672 sectors), 612 cyls, 8 heads, 32 S/T, 512 B/S
wdc0: unit 1 (wd1): <IBM-DCXA-210000>
wd1: 8063MB (16514064 sectors), 16383 cyls, 16 heads, 63 S/T, 512 B/S
wdc1 not found at 0x170
wdc2 not found at 0x180
ep0 not found at 0x300
fxp0: Ethernet address 00:a0:a5:12:05:5a
fxp1: Ethernet address 00:a0:a5:12:05:59
fxp2: Ethernet address 02:00:00:00:00:01
swapon: adding /dev/wd1s1b as swap device
Automatic reboot in progress...
/dev/rwd0s1a: clean, 16599 free (95 frags, 2063 blocks, 0.1% fragmentation)
/dev/rwd0s1e: clean, 9233 free (9 frags, 1153 blocks, 0.1% fragmentation)
/dev/rwd0s1a: clean, 16599 free (95 frags, 2063 blocks, 0.1% fragmentation)
/dev/rwd1s1f: clean, 4301055 free (335 frags, 537590 blocks, 0.0% fragmentation)

```

Meaning

The sample output shows the initial messages generated by the system kernel upon boot. This is the content of the `/var/run/dmesg.boot` file.

Log the Active Configuration (Junos OS)

IN THIS SECTION

- [Action | 58](#)
- [Sample Output | 58](#)
- [Meaning | 58](#)

Action

To log the active configuration on the router, use the following Junos OS CLI operational mode command:

```
user@host> show configuration | save filename
```

Sample Output

```
user@host> show configuration | save test
Wrote 4076 lines of output to 'test'

user@host> show configuration
system {
  host-name lab8;
  domain-name device1.example.com;
  backup-router 10.1.1.254;
    time-zone America/Los_Angeles;
  default-address-selection;
    dump-on-panic;
  name-server {
  [...Output truncated...]
```

Meaning

The sample output shows the configuration currently running on the router, which is the last committed configuration.

Log the Interfaces on the Router (Junos OS)

IN THIS SECTION

- [Action | 59](#)
- [Sample Output | 59](#)
- [Meaning | 59](#)

Action

To log the interfaces on the router, use the following Junos OS CLI operational mode command:

```
user@host> show interface terse | save filename
```

Sample Output

```
user@host> show interfaces terse | save test
Wrote 81 lines of output to 'test'

user@host> show interfaces terse
Interface      Admin Link Proto Local          Remote
at-1/3/0       up    up
at-1/3/0.0     up    up    inet 203.0.113.1    --> 203.0.113.2
               iso
fxp0           up    up
fxp0.0         up    up    inet 10.168.5.59/24
gre            down  up
ipip           down  up
lo0            up    up
lo0.0          up    up    inet 127.0.0.1        --> 0/0
               iso 47.0005.80ff.f800.0000.0108.0001.1921.6800.5059.00
so-1/2/0       up    down
so-1/2/1       down  down
so-1/2/2       down  down
so-1/2/3       down  down
so-2/0/0       up    up
so-2/0/0.0     up    up    inet 192.2.3.4        --> 192.2.3.5
               iso
[...Output truncated...]
```

Meaning

The sample output displays summary information about the physical and logical interfaces on the router.

Log the BGP, IS-IS, and OSPF Adjacency Information (Junos OS)

IN THIS SECTION

- Purpose | 60
- Action | 60
- Sample Output 1 | 60
- Sample Output 2 | 61
- Sample Output 3 | 61
- Meaning | 62

Purpose

The following commands log useful information about Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), and Open Shortest Path First (OSPF) protocols. If you have other protocols installed, such as Multiprotocol Label Switching (MPLS), Resource Reservation Protocol (RSVP), or Protocol Independent Multicast (PIM), you also might log summary information for them.

Action

To log the protocol peer information, use the following Junos OS CLI operational mode commands:

```
user@host> show bgp summary | save filename
user@host> show isis adjacency brief | save filename
user@host> show ospf neighbor brief | save filename
```

Sample Output 1

```
user@host> show bgp summary | save test
Wrote 45 lines of output to 'test'

user@host> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State   Pending
inet.0         4          4          0          0        0        0
Peer           AS        InPkt   OutPkt   OutQ   Flaps Last Up/Dwn State|#Active/Received/
```

```
Damped..
9.9.3.1          2      2627      2628      0      0      21:50:12 4/4/0
0/0/0
```

Sample Output 2

```
user@host> show isis adjacency brief | save test
Wrote 7 lines of output to 'test'
```

```
user@host> show isis adjacency brief
IS-IS adjacency database:
Interface System      L State      Hold (secs) SNPA
so-1/0/0.0 1921.6800.5067 2 Up          13
so-1/1/0.0 1921.6800.5067 2 Up          25
so-1/2/0.0 1921.6800.5067 2 Up          20
so-1/3/0.0 1921.6800.5067 2 Up          19
so-2/0/0.0 1921.6800.5066 2 Up          19
so-2/1/0.0 1921.6800.5066 2 Up          17
so-2/2/0.0 1921.6800.5066 2 Up          20
so-2/3/0.0 1921.6800.5066 2 Up          20
so-5/0/0.0 ranier      2 Up          17
```

Sample Output 3

```
user@host> show ospf neighbor brief | save test
Wrote 10 lines of output to 'test'
```

```
user@host> show ospf neighbor brief
Address      Intf      State      ID          Pri  Dead
10.168.254.225 fxp3.0    2Way       10.250.240.32 128  36
10.168.254.230 fxp3.0    Full       10.250.240.8  128  38
10.168.254.229 fxp3.0    Full       10.250.240.35 128  33
10.1.1.129      fxp2.0    Full       10.250.240.12 128  37
10.1.1.131      fxp2.0    Full       10.250.240.11 128  38
10.1.2.1        fxp1.0    Full       10.250.240.9  128  32
10.1.2.81       fxp0.0    Full       10.250.240.10 128  33
```

Meaning

Sample output 1 displays summary information about BGP and its neighbors. Sample output 2 displays information about IS-IS neighbors. Sample output 3 displays information about all OSPF neighbors.

Log the System Storage Information (Junos OS)

IN THIS SECTION

- [Action | 62](#)
- [Sample Output | 62](#)
- [Meaning | 63](#)

Action

To log the system storage statistics for the amount of free disk space in the router's file system, use the following Junos OS CLI operational mode command:

```
user@host> show system storage | save filename
```

Sample Output

```
user@host> show system storage | save test
Wrote 14 lines of output to 'test'
```

```
user@host> show system storage
```

Filesystem	1K-blocks	Used	Avail	Capacity	Mounted on
/dev/ad0s1a	65687	26700	33733	44%	/
devfs	16	16	0	100%	/dev/
/dev/vn1	9310	9310	0	100%	/packages/mnt/jbase
/dev/vn2	8442	8442	0	100%	/packages/mnt/jkernel-5.0R5.1
/dev/vn3	11486	11486	0	100%	/packages/mnt/jpfe-5.0R5.1
/dev/vn4	5742	5742	0	100%	/packages/mnt/jroute-5.0R5.1
/dev/vn5	1488	1488	0	100%	/packages/mnt/jcrypto-5.0R5.1
/dev/vn6	792	792	0	100%	/packages/mnt/jdocs-5.0R5.1
mfs:2373	1015815	3	934547	0%	/tmp
/dev/ad0s1e	25263	11	23231	0%	/config

procfs	4	4	0	100%	/proc
/dev/ad1s1f	9825963	1811085	7228801	20%	/var

Meaning

The sample output displays statistics about the amount of free disk space in the router's file system. Values are displayed in 1024-byte (1-KB) blocks.

Validating the Configuration Image Before Upgrading or Downgrading the Software (Junos OS)

Here are some validation guidelines to keep in mind:

- Validation is set to on by default. You do not need to configure it or issue any command to start it on a switch that supports image validation. You can disable validation (the procedure is given below) and then re-enable it.
- Validation slows down the upgrade or downgrade process by as much as 7 minutes.
- Image validation is supported only on the **jinstall** package.
- If you invoke validation from an image that does not support validation, the new image is loaded but validation does not occur.
- Validation does not work in a *downgrade* to an image that does not support validation if your system is configured for graceful routing switchover (GRES) or if you run image loading without nonstop software upgrade (NSSU). See the procedure below for steps to use validation in this type of scenario.

If you upgrade or downgrade the Junos OS image on a switch that supports configuration image validation (see [Feature Explorer](#) for feature support per EX Series switch), the system validates that the existing configuration is compatible with the new image before the actual upgrade or downgrade commences.

Benefits of image validation—If validation fails, the new image is not loaded, and an error message provides information about the failure. If you upgrade or downgrade the software on a system that does not support validation, configuration incompatibilities between the existing and new image or insufficient memory to load the new image might cause the system to lose its current configuration or go offline.

To disable validation, re-enable or invoke validation manually, or use validation when downgrading to an image that does not support it:

- To disable validation, issue request system software add *image-name* reboot no-validate command.
- To re-enable or invoke validation manually, choose one of the following methods:

- Issue `request system software add image-name`.
- Issue `request system software nonstop-upgrade image-name`.
- Issue `request system software validate` to run just configuration validation.
- To use validation when downgrading to an image that does not support it, choose one of the following methods:
 - Remove the graceful-switchover configuration and then issue the `request system software add image-name reboot` command.
 - Use NSSU by issuing the `request system software nonstop-upgrade image-name` command.

Ensure Sufficient Disk Space for Junos OS Upgrades on SRX Series Firewalls

IN THIS SECTION

- [Verify Available Disk Space on SRX Series Devices | 64](#)
- [Cleanup the System File Storage Space | 65](#)

Before you begin upgrading Junos OS on an SRX Series Firewall, perform the following tasks:

Verify Available Disk Space on SRX Series Devices

The amount of free disk space necessary to upgrade a device with a new version of Junos OS can vary from one release to another. Check the Junos OS software version you are installing to determine the free disk space requirements.

If the amount of free disk space on a device is insufficient for installing Junos OS, you might receive a warning similar to the following messages, that the `/var` filesystem is low on free disk space:

WARNING: The /var filesystem is low on free disk space.

WARNING: This package requires 1075136k free, but there is only 666502k available.

To determine the amount of free disk space on the device, issue the `show system storage detail` command. The command output displays statistics about the amount of free disk space in the device file systems.

A sample of the `show system storage detail` command output is shown below:

```
user@host> show system storage detail
```


Filesystem	1024-blocks	Used	Avail	Capacity	Mounted on
/dev/da0s2a	300196	154410	121772	56%	/
devfs	1	1	0	100%	/dev
/dev/md0	409000	409000	0	100%	/junos
/cf	300196	154410	121772	56%	/junos/cf
devfs	1	1	0	100%	/junos/dev/
procfs	4	4	0	100%	/proc
/dev/bo0s3e	25004	52	22952	0%	/config
/dev/bo0s3f	350628	178450	144128	55%	/cf/var
/dev/md1	171860	16804	141308	11%	/mfs
/cf/var/jail	350628	178450	144128	55%	/jail/var
/cf/var/log	350628	178450	144128	55%	/jail/var/log
devfs	1	1	0	100%	/jail/dev
/dev/md2	40172	4	36956	0%	/mfs/var/run/utm
/dev/md3	1884	138	1596	8%	/jail/mfs

Cleanup the System File Storage Space

When the system file storage space on the device is full, rebooting the device does not solve the problem. The following error message is displayed during a typical operation on the device after the file storage space is full.

```
user@host% cli
user@host> configure/var: write failed, filesystem is full
```

You can clean up the file storage on the device by deleting system files using the `request system storage cleanup` command as shown in following procedure:

1. Request to delete system files on the device.

```
user@host> request system storage cleanup
```

The list of files to be deleted is displayed.

List of files to delete:

Size	Date	Name
11B	Oct 28 23:40	/var/jail/tmp/alarmd.ts
92.4K	Jan 11 17:12	/var/log/chassisd.0.gz
92.4K	Jan 11 06:06	/var/log/chassisd.1.gz

```

92.5K Jan 10 19:00 /var/log/chassisd.2.gz
92.5K Jan 10 07:53 /var/log/chassisd.3.gz
92.2K Jan 10 15:00 /var/log/hostlogs/auth.log.1.gz
92.2K Jan 1 18:45 /var/log/hostlogs/auth.log.2.gz
92.1K Jan 4 17:30 /var/log/hostlogs/auth.log.3.gz
92.2K Jan 1 18:45 /var/log/hostlogs/auth.log.4.gz
79.0K Jan 12 01:59 /var/log/hostlogs/daemon.log.1.gz
78.8K Jan 11 23:15 /var/log/hostlogs/daemon.log.2.gz
78.7K Jan 11 20:30 /var/log/hostlogs/daemon.log.3.gz
79.1K Jan 11 17:44 /var/log/hostlogs/daemon.log.4.gz
59.1K Jan 11 21:59 /var/log/hostlogs/debug.1.gz
59.2K Jan 11 17:44 /var/log/hostlogs/debug.2.gz
59.2K Jan 11 13:29 /var/log/hostlogs/debug.3.gz
59.3K Jan 11 09:14 /var/log/hostlogs/debug.4.gz
186.6K Oct 20 16:31 /var/log/hostlogs/kern.log.1.gz
238.3K Jan 11 23:15 /var/log/hostlogs/lcmd.log.1.gz
238.4K Jan 11 17:30 /var/log/hostlogs/lcmd.log.2.gz
238.6K Jan 11 11:45 /var/log/hostlogs/lcmd.log.3.gz
238.5K Jan 11 06:00 /var/log/hostlogs/lcmd.log.4.gz
372.5K Jan 11 17:00 /var/log/hostlogs/syslog.1.gz
372.5K Jan 11 04:45 /var/log/hostlogs/syslog.2.gz
371.9K Jan 10 16:30 /var/log/hostlogs/syslog.3.gz
372.7K Jan 10 04:15 /var/log/hostlogs/syslog.4.gz
10.1K Jan 12 02:03 /var/log/messages.0.gz
55.1K Jan 6 21:25 /var/log/messages.1.gz
81.5K Dec 1 21:30 /var/log/messages.2.gz

```

Delete these files ? [yes,no] (no)

2. Enter the option yes to proceed with deleting of the files.

Verifying Junos OS and Boot Loader Software Versions on an EX Series Switch

IN THIS SECTION

- [Verifying the Number of Partitions and File System Mountings | 67](#)
- [Verifying the Loader Software Version | 68](#)
- [Verifying Which Root Partition Is Active | 70](#)
- [Verifying the Junos OS Version in Each Root Partition | 71](#)

Before or after upgrading or downgrading Junos OS, you might need to verify the Junos OS version. You might also need to verify the boot loader software version if you are upgrading to or downgrading from a release that supports resilient dual-root partitions (Junos OS Release 10.4R3 and later).

This topic includes:

Verifying the Number of Partitions and File System Mountings

IN THIS SECTION

- Purpose | 67
- Action | 67
- Meaning | 68

Purpose

Between Junos OS Release 10.4R2 and Release 10.4R3, upgrades were made to further increase resiliency of root partitions, which required reformatting the disk from three partitions to four partitions. If your switch is running Release 10.4R2 or earlier, it has three partitions, and if it is running Release 10.4R3 or later, it has four partitions.

Action

Verify how many partitions the disk has, as well as where each file system is mounted, by using the following command:

```
user@switch> show system storage
fpc0:
-----
Filesystem  Size  Used  Avail  Capacity  Mounted on
/dev/da0s1a 184M 124M   45M    73%      /
devfs       1.0K 1.0K   0B     100%    /dev
/dev/md0     37M  37M   0B     100%    /packages/mnt/jbase
/dev/md1     18M  18M   0B     100%    /packages/mnt/jcrypto-
ex-10.4I20110121_0509_hbRPSRLI15184421081
/dev/md2     6.1M 6.1M   0B     100%    /packages/mnt/jdocs-
ex-10.4I20110121_0509_hbRPSRLI15184421081
/dev/md3    154M 154M   0B     100%    /packages/mnt/jkernel-
```

```

ex-10.4I20110121_0509_hbRPSRLI15184421081
/dev/md4      23M   23M   0B      100% /packages/mnt/jpfe-
ex42x-10.4I20110121_0509_hbRPSRLI15184421081
/dev/md5      46M   46M   0B      100% /packages/mnt/jroute-
ex-10.4I20110121_0509_hbRPSRLI15184421081
/dev/md6      28M   28M   0B      100% /packages/mnt/jswitch-
ex-10.4I20110121_0509_hbRPSRLI15184421081
/dev/md7      22M   22M   0B      100% /packages/mnt/jweb-
ex-10.4I20110121_0509_hbRPSRLI15184421081
/dev/md8      126M  10.0K 116M     0% /tmp
/dev/da0s3e   123M   632K 112M     1% /var
/dev/da0s3d   369M    20K 339M     0% /var/tmp
/dev/da0s4d    62M    62K  57M     0% /config
/dev/md9      118M   12M   96M    11% /var/rundb
procfs        4.0K   4.0K   0B     100% /proc
/var/jail/etc 123M   632K 112M     1% /packages/mnt/jweb-
ex-10.4I20110121_0509_hbRPSRLI15184421081/jail/var/etc
/var/jail/run 123M   632K 112M     1% /packages/mnt/jweb-
ex-10.4I20110121_0509_hbRPSRLI15184421081/jail/var/run
/var/jail/tmp 123M   632K 112M     1% /packages/mnt/jweb-
ex-10.4I20110121_0509_hbRPSRLI15184421081/jail/var/tmp
/var/tmp      369M    20K 339M     0% /packages/mnt/jweb-
ex-10.4I20110121_0509_hbRPSRLI15184421081/jail/var/tmp/uploads
devfs         1.0K   1.0K   0B     100% /packages/mnt/jweb-
ex-10.4I20110121_0509_hbRPSRLI15184421081/jail/dev

```

Meaning

The presence of the partition name containing s4d indicates that there is a fourth slice. If this were a three-slice partition scheme, in place of s1a, s3e, s3d, and s4d, you would see s1a, s1f, s2a, s2f, s3d, and s3e, and you would not see s4d.

Verifying the Loader Software Version

IN THIS SECTION

- Purpose | 69
- Action | 69
- Meaning | 69

Purpose

For the special case of upgrading from Junos OS Release 10.4R2 or earlier to Release 10.4R3 or later, you must upgrade the loader software.

Action

For EX Series switches except EX8200 switches:

```
user@switch> show chassis firmware
```

Part	Type	Version
FPC 0	uboot	U-Boot 1.1.6 (Jan 3 2011 - 16:14:58) 1.0.0
	loader	FreeBSD/PowerPC U-Boot bootstrap loader 2.4

For EX8200 switches:

```
user@switch> show chassis firmware
```

Part	Type	Version
FPC 0	uboot	U-Boot 1.1.6 (Jan 3 2011 - 16:14:58) 3.5.0
	loader	FreeBSD/PowerPC U-Boot bootstrap loader 2.4

Meaning

For EX Series switches other than EX8200 switches, with Junos OS Release 10.4R3 or later installed:

- If there is version information following the timestamp for U-Boot (1.0.0 in the preceding example), then the loader software does not require upgrading.
- If there is no version number following the timestamp for U-boot, then the loader software requires upgrading.



NOTE: If the software version is Release 10.4R2 or earlier, no version number is displayed following the timestamp for U-boot, regardless of the loader software version installed. If you do not know whether you have installed the new loader software, we recommend that you upgrade the loader software when you upgrade the software version.

For EX8200 switches, if the version number following the timestamp for U-Boot is earlier than 3.5.0, you must upgrade the loader software when you upgrade the software version.

Verifying Which Root Partition Is Active

IN THIS SECTION

- Purpose | 70
- Action | 70
- Meaning | 70

Purpose

Switches running Release 10.4R3 or later have resilient dual-root partition functionality, which includes the ability to boot transparently from the inactive partition if the system fails to boot from the primary root partition.

You can verify which root partition is active using the following command:

Action

```
user@switch> show system storage partitions
```

```
fpc0:
```

```
-----
```

```
Boot Media: internal (da0)
```

```
Active Partition: da0s1a
```

```
Backup Partition: da0s2a
```

```
Currently booted from: active (da0s1a)
```

```
Partitions information:
```

Partition	Size	Mountpoint
s1a	184M	/
s2a	184M	altroot
s3d	369M	/var/tmp
s3e	123M	/var
s4d	62M	/config
s4e		unused (backup config)

Meaning

The Currently booted from: field shows which root partition is active.

Verifying the Junos OS Version in Each Root Partition

IN THIS SECTION

- Purpose | 71
- Action | 71
- Meaning | 72

Purpose

Each switch contains two root partitions. We recommend that you copy the same Junos OS version in each partition when you upgrade. In Junos OS Release 10.4R2 and earlier, you might choose to have different Junos OS release versions in each partition. You might have different versions during a software upgrade and before you have finished verifying the new software installation. To enable a smooth reboot if corruption is found in the primary root file system, ensure that the identical Junos OS images are in each root partition. For Release 10.4R2 and earlier, you must manually reboot the switch from the backup root partition. However, for Release 10.4R3 and later, the switch reboots automatically from the backup root partition if it fails to reboot from the active root partition.

Action

Verify whether both root partitions contain the same image by using the following command:

```
user@switch> show system snapshot media internal
Information for snapshot on      internal (/dev/da0s1a) (backup)
Creation date: Jan 11 03:02:59 2012
JUNOS version on snapshot:
  jbase   : ex-12.2I20120305_2240_user
  jcrypto-ex: 12.2I20120305_2240_user
  jdocs-ex: 12.2I20120305_2240_user
  jroute-ex: 12.2I20120305_2240_user
  jswitch-ex: 12.2I20120305_2240_user
  jweb-ex: 12.2I20120305_2240_user
Information for snapshot on      internal (/dev/da0s2a) (primary)
Creation date: Mar 6 02:24:08 2012
JUNOS version on snapshot:
  jbase   : ex-12.2I20120305_2240_user
  jcrypto-ex: 12.2I20120305_2240_user
```

```

jdocs-ex: 12.2I20120305_2240_user
jroute-ex: 12.2I20120305_2240_user
jswitch-ex: 12.2I20120305_2240_user
jweb-ex: 12.2I20120305_2240_user

```

Meaning

The command shows which Junos OS version is installed on each media partition. Verify that the same version is installed on both partitions.

RELATED DOCUMENTATION

| [Configuring Dual-Root Partitions](#) | 161

Access Juniper Support

IN THIS SECTION

- [Existing Users—How to Log In](#) | 72
- [New Users—How to Create an Account](#) | 72

This topic provides an overview on how you can access the software package downloads and support tools.

Existing Users—How to Log In

If you are an existing user with an active Juniper Networks® profile, contact [Global support](#). The global support team sends an access token to your registered e-mail ID.

New Users—How to Create an Account

To register as a new user, click the [User Registration](#) link and perform the following steps to create a new account:

1. Create a user account by providing your e-mail address on the [Create User Account](#) page.
After you submit your e-mail ID, you will receive a confirmation e-mail with a link to proceed with the account setup process.
2. Click the link to open the **Account Setup** page and complete all the required account setup activities.

The **Email Address** field already contains the e-mail address you provided in Step 1. This e-mail ID also acts as your user ID for this account.



NOTE: You cannot create an account by using a public domain e-mail address such as @gmail.com or @yahoo.com. If you use a public domain address, you will receive an alert declining your account status. Change your e-mail address before you click **Next** to proceed.

3. If you are not an existing Juniper customer or partner and the system does not recognize your email domain, you can select one of the following options:

- Individual Email
- Group Email

Hover over the question mark icons next to each option for a brief description.

4. Click **Next** to proceed.

Your account creation is successful.

5. After your account is active, contact [Global support](#). The global support team sends an access token to your registered e-mail ID.

SEE ALSO

<https://www.juniper.net/documentation/us/en/software/crpd/crpd-deployment/topics/task/crpd-linux-server-install.html>

Downloading Software (Junos OS)

IN THIS SECTION

- [Downloading Software Using a Browser \(Junos OS\) | 73](#)
- [Downloading Software Using the Command-Line Interface \(Junos OS\) | 75](#)
- [Download Software Using Download Manager \(SRX Series Only\) | 77](#)

Downloading Software Using a Browser (Junos OS)

You download the software package you need from the Juniper Networks Downloads page at <https://support.juniper.net/support/downloads/>.



NOTE: To access the download section, you must have a service contract and an access account. If you require assistance in acquiring an account, refer to the instructions on how to "Access Juniper Support" on page 72 and fill out the registration form found on the Juniper Networks website: <https://userregistration.juniper.net/entitlement/setupAccountInfo.do>.

To download the software image:

1. Using a Web browser, navigate to <https://support.juniper.net/support/downloads/>.

The Download Results page appears.

2. Find the software package that you want to download and click the item in the Downloads column.

A login screen appears.

3. Log in with your username and password.

4. On the Download Software page that appears, the following options are available:

- If you want to download the software on your local host, click the **CLICK HERE** link and save the file to your system. If you want to place the file on a remote system, you must make sure that the file can be accessible by the router, switch, or services gateway by using HTTP, FTP, or SCP. Proceed with the installation. See "Downloading Software (Junos OS)" on page 73 for more details.
- If you want to download the software on your device, use the following procedure to download and install the software on the device.
 - a. Click **Copy** to copy the generated URL to the clipboard.



NOTE: The URL string generated remains active only for 15 minutes.

- b. Log in to your device.
- c. In operational mode, enter the file copy "*URL*" *destination* command.

In the command, paste the copied URL string (for *URL*) and then enter */var/tmp* (as the destination on your hard disk).

Example:

```
user@host> file copy "URL" /var/tmp
```



NOTE: Ensure that the URL string is enclosed within quotation marks. Also ensure that there is sufficient free space available on the device.

The software image is downloaded on your device.

- d. (Optional) Validate the software image by using the `request system software validate package-name` command.

Example:

```
user@host> request system software validate /var/tmp/junos-install-mx-
x86-32-17.3R1.10.tgz
```

For more details, see [request system software validate](#).

- e. Install the software by using the `request system software add package-name` command.

Example:

```
user@host> request system software add /var/tmp/junos-install-mx-x86-32-17.3R1.10.tgz
```

Your software is installed on the device.

Downloading Software Using the Command-Line Interface (Junos OS)

Download the software package you need from the Juniper Networks Downloads page at <https://support.juniper.net/support/downloads/>, and place the package on a local system. You can then transfer the downloaded package to the device using either the router or switch command-line interface, or the local system command-line interface.



NOTE: To access the download section, you must have a service contract and an access account. If you need help obtaining an account, complete the registration form at the Juniper Networks website: <https://userregistration.juniper.net/entitlement/setupAccountInfo.do>.

Before you transfer the software package, ensure that the FTP service is enabled on the device. Enable the FTP service using the `set system services ftp` command:

```
user@host# set system services ftp
```

To transfer the software package using the device command-line interface:

1. From the router or switch command line, initiate an FTP session with the local system (host) where the package is located by using the `ftp` command:

```
user@host> ftp host
```

host is the hostname or address of the local system.

2. Log in with your customer support-supplied username and password:

```
User Name: username  
331 Password required for username.  
Password: password
```

After your credentials are validated, the FTP session opens.

3. Navigate to the software package location on the local system, and transfer the package by using the `get` command:

```
user@host> get installation-package
```

Following is an example of an *installation-package* name: **junos-install-mx-x86-32-17.3R1.10.tgz**

4. Close the FTP session by using the `bye` command:

```
user@host> bye  
Goodbye
```

To transfer the package by using the local system command-line interface:

1. From the local system command line, initiate an FTP session with the device using the `ftp` command:

```
user@host> ftp host
```

host is the hostname or address of the router or switch.

2. Log in with your customer support-supplied username and password:

```
User Name: username
331 Password required for username.
Password: password
```

After your credentials are validated, the FTP session opens.

3. Navigate to the software package location on the local system, and transfer the package by using the put command:

```
user@host> put installation-package
```

Following is an example of an *installation-package* name: **junos-install-mx-x86-32-17.3R1.10.tgz**

4. Close the FTP session by using the bye command:

```
user@host> bye
Goodbye
```

Download Software Using Download Manager (SRX Series Only)

This download manager feature facilitates download of large files over low-bandwidth links. It enables you to download large Junos OS packages over low-bandwidth/flaky links so that the system can be upgraded. This feature allows you to download multiple files while monitoring their status and progress individually. It takes automatic action when required and displays status information when requested.

The download manager is supported on SRX300, SRX320, SRX340, SRX345, and SRX380 devices.

Be aware of the following considerations when using the download manager:

- When no download limit is specified for a specific download or for all downloads, a download uses all available network bandwidth.
- Because the download limit that you set indicates an average bandwidth limit, it is possible that certain bursts might exceed the specified limit.
- When a download from an HTTP server fails, the server returns an HTML page. Occasionally, the error page is not recognized as an error page and is downloaded in place of the Junos image file.
- Remote server logins and passwords are stored by the download manager for the duration of a download. To encrypt these credentials provided along with the login keyword, define an encryption

key with the `request system set-encryption-key` command. Any changes to encryption settings while download is in progress can cause the download to fail.

- A download command issued on a particular node in a *chassis cluster* takes place only on that node and is not propagated to the other nodes in the cluster. Downloads on different nodes are completely independent of each other. In the event of a failover, a download continues only if the server remains reachable from the node from which the command was issued. If the server is no longer reachable on that node, the download stops and returns an error.

The download manager supports only the FTP and HTTP protocols.

The download manager acts as a substitute for the FTP utility. You can use the download manager CLI commands for all the functions where you previously used the FTP utility.

Before you begin, you must have the following:

- An FTP or HTTP server with a Junos OS image
- A server that is reachable from the device being upgraded

To download the Junos OS image to your device:

1. Use the `request system download start` command (set a bandwidth limit, if required). The file is saved to the `/var/tmp` directory on your device.

You can continue to use the device while the download runs in the background.

2. To verify that the file has been downloaded, use the `show system download` command. The command displays the state as "completed" when the downloaded file is ready to be installed.
3. To install the downloaded image file from the `/var/tmp` directory, use the `request system software add` command.
4. If you encounter any problem with a download, use the `show system download id` command to obtain details about the download.

Table 2 lists the output fields for the `show system download` command. Use this information to diagnose problems. Output fields are listed in the approximate order in which they appear.

Table 12: show system download Output Fields

Output Field	Description
Status	State of the download.
Creation Time	Time the start command was issued.

Table 12: show system download Output Fields (Continued)

Output Field	Description
Scheduled Time	Time the download was scheduled to start.
Start Time	Time the download actually started (if it has already started).
Retry Time	Time for next retry (if the download is in the error state).
Error Count	Number of times an error was encountered by this download.
Retries Left	Number of times the system will retry the download automatically before stopping.
Most Recent Error	Message indicating the cause of the most recent error.

Reinstall Junos OS

IN THIS SECTION

- [Action | 79](#)

Action

To reinstall Junos OS, follow these steps:

1. Insert the removable medium (boot floppy) into the router.
2. Reboot the router, either by power-cycling it or by issuing the request `system reboot` command from the CLI.
3. At the following prompt, type `y`:

```
WARNING: The installation will erase the contents of your disk. Do you wish to continue (y/n)?
```

The router copies the software from the removable medium onto your system, occasionally displaying status messages. This can take up to 10 minutes.

4. Remove the removable medium when prompted.

The router reboots from the primary boot device on which the software is installed. When the reboot is complete, the router displays the login prompt.

Reconfigure Junos OS

IN THIS SECTION

- [Configure Host Names, Domain Names, and IP Addresses \(Junos OS\) | 80](#)
- [Protect Network Security by Configuring the Root Password | 81](#)
- [Check Network Connectivity \(Junos OS\) | 83](#)

After you have reinstalled the software, you must copy the router's configuration files back to the router. (You also can configure the router from scratch, as described in *Junos System Basics Configuration Guide*) However, before you can copy the configuration files, you must establish network connectivity.

To reconfigure the software, follow these steps:

Configure Host Names, Domain Names, and IP Addresses (Junos OS)

To configure the machine name, domain name, and various addresses, follow these steps:

1. Log in as root. There is no password.
2. Start the CLI:

```
root# cli
root@>
```

3. Enter configuration mode:

```
root@> configure
[edit]
root@#
```


4. Configure the name of the machine. If the name includes spaces, enclose the entire name in quotation marks (" "):

```
[edit]
root@# set system host-name host-name
```

5. Configure the machine's domain name:

```
[edit]
root@# set system domain-name domain-name
```

6. Configure the IP address and prefix length for the router's management Ethernet interface:

```
[edit]
root@# set interfaces fxp0 unit 0 family inet address address | prefix-length
```

7. Configure the IP address of a default router. This system is called the backup router because it is used only while the routing protocol process is not running.

```
[edit]
root@# set system backup-router address
```

8. Configure the IP address of a Domain Name Server (DNS) server:

```
[edit]
root@# set system name-server address
```

Protect Network Security by Configuring the Root Password

Configure the root password on your Juniper Networks device to help prevent unauthorized users from making changes to your network. The root user (also referred to as superuser) has unrestricted access and full permissions within the system, so it is crucial that you protect this account by setting a strong password when setting up a new device.

After you initially power on a new device, you log in as the user root with no password. The software requires you to configure the root password before it accepts a commit operation.

To set the root password, you have three options:

- Enter a plain-text password that the software encrypts.
- Enter a password that is already encrypted.
- Enter a Secure Shell (SSH) public key string.

Among these options, using a pre-encrypted password or an SSH public key string is the most secure. If you use one of these methods, then the plain-text version of your password will never be transferred over the Internet, protecting it from being intercepted by a man-in-the-middle attack.



BEST PRACTICE: Optionally, instead of configuring the root password at the [edit system] hierarchy level, you can use a configuration group to strengthen security.

To set the root password:

1. Use one of these methods to configure the root password:

- To enter a plain-text password that the system encrypts for you:

```
[edit groups global system]
root@# set root-authentication plain-text-password
New Password: type password here
Retype new password: retry password here
```

As you enter a plain-text password into the CLI, the device software hides it from view and encrypts it immediately. You don't have to configure the software to encrypt the password. In the resulting configuration, the encrypted password is marked as ## SECRET-DATA so that it cannot be seen.

- To enter a password that is already encrypted:



CAUTION: Do not use the encrypted-password option unless the password is *already* encrypted and you are entering that encrypted password.

If you accidentally configure the encrypted-password option with a plain-text password or with blank quotation marks (" "), you will not be able to log in to the device as the root user. You will then need to complete the root password recovery process.

```
[edit groups global system]
root@# set root-authentication encrypted-password password
```

- To enter an SSH public key string:

```
[edit groups global system]
root@# set root-authentication (ssh-ecdsa | ssh-rsa key)
```

2. If you used a configuration group, replace the *group-name* variable with the configuration group's name.

```
[edit]
root@# set apply-groups group-name
```

3. Commit the changes.

```
root@# commit
```

Check Network Connectivity (Junos OS)

IN THIS SECTION

- [Purpose | 83](#)
- [Action | 83](#)

Purpose

Establish that the router has network connectivity.

Action

To check that the router has network connectivity, issue a `ping` command to a system on the network:

```
root@> ping address
```

If there is no response, verify that there is a route to the *address* using the `show route` command. If the address is outside your `fxp0` subnet, add a static route. Once the backup configuration is loaded and committed, the static route is no longer needed and should be deleted.

Change History Table

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

Release	Description
18.3R1	Starting in Junos OS Release 18.3R1, the ssh-dss and ssh-dsa hostkey algorithms are deprecated—rather than immediately removed—to provide backward compatibility and a chance to bring your configuration into compliance with the new configuration.

RELATED DOCUMENTATION

[Troubleshooting Software Installation on EX Series Switches | 165](#)

[Troubleshooting a Switch That Has Booted from the Backup Junos OS Image | 170](#)

Managing Disk Space for Junos OS Installation

A Junos OS installation or upgrade may fail if your router has a shortage of disk space. If a disk space error occurs, use one or more of the following options to complete the installation:

- Use the `request system storage cleanup` command to delete unnecessary files and increase storage space on the router.
- Specify the `unlink` option when you use the `request system software add` command to install the Junos OS:
 - On the M Series, MX Series, and T Series routers, the `unlink` option removes the software package after a successful upgrade.
- Download the software packages you need from the Juniper Networks Support Web site at <https://www.juniper.net/support/>. The download program provides intelligent disk space management to enable installation.

RELATED DOCUMENTATION

[Junos OS Configuration Using the CLI](#)

Verifying PIC Combinations (Junos OS)

On Juniper Networks routing platforms, you can typically install any combination of Physical Interface Cards (PICs) on a single Enhanced Flexible PIC Concentrator (FPC) or in two PIC slots served by a single Layer 2/Layer 3 Packet Processing application-specific integrated circuit (ASIC).

Newer Junos OS services for some PICs can require significant Internet Processor ASIC memory, and some configuration rules limit certain combinations of PICs if they are installed on some platforms.

During software installation, the configuration checker in the installation program checks the router's PICs. If any configuration rules affect your PIC combinations, the installation process stops and displays a message similar to the following:

```
The combination of PICS in FPC slot 3 is not supported with this release
PIC slot 0 -
PIC slot 1 - 1x OC-12 ATM-II IQ
PIC slot 2 - 1x G/E IQ, 1000 BASE
PIC slot 3 - 1x Link Service (4)
If you continue the installation, one or more PICs on
FPC slot 3 might appear to be online but
cannot be enabled and cannot pass traffic with this release of JUNOS.
See the Release Notes for more information.
WARNING: This installation attempt will be aborted. If you
WARNING: wish to force the installation despite these warnings
WARNING: you may use the 'force' option on the command line.
pkg_add: package /var/tmp/jbundle-7.6R1.x-domestic-signed.tgz fails requirements - not installed
```

The configuration checker has the following limitations:

- If a PIC is offline when you upgrade the router with new software, the configuration checker cannot detect PIC combinations affected by configuration rules and cannot warn about them.
- If you specify the **force** option when you upgrade the Junos OS, the configuration checker warns about the affected PIC combination and the software installation continues. However, after rebooting, one or more PICs might fail to initialize.
- The configuration checker looks for combinations of three affected PICs. If an Enhanced FPC contains four affected PICs, the script generates multiple warnings.

If you install a PIC into a router already running Junos OS, you can identify the presence of affected PIC combinations from messages in the system logging (**syslog**) file:

```
Feb 6 17:57:40 CE1 feb BCHIP 0: uCode overflow - needs 129 inst space to load
b3_atm2_LSI_decode for stream 12
Feb 6 17:57:41 CE1 chassisd[2314]: CHASSISD_IFDEV_DETACH_PIC: ifdev_detach_pic(0/3)
Feb 6 17:57:41 CE1 feb BCHIP 0: binding b3_atm2_LSI_decode to stream 12 failed
Feb 6 17:57:41 CE1 feb PFE: can not bind B3 ucode prog b3_atm2_LSI_decode to FPC 0: stream 12
```

For more information about checking for unsupported PIC combinations, see the corresponding PIC guide for your router, the [Junos OS Release Notes](#), and *Technical Support Bulletin PSN-2004-12-002, PIC Combination Notes Summary* on the Juniper Networks Support Web site at <https://www.juniper.net/support/>.

For SRX Series Services Gateways

SRX5600 and SRX5800 devices support IOC or SPC on any given card slot, and there is no complexity in equipping the services gateways with the perfect balance of processing and I/O capacity. You can install up to 11 (on SRX5800) and 5 (SRX5600) SPCs and IOCs on the device. However, you must install at least one SPC on device. For more details, see [SRX5600 and SRX5800 Services Gateway Card Guide](#).

SRX3600 supports a maximum of up to seven SPCs, three NPCs, six IOCs, and 11 NP-IOCs per chassis. However you must install at least one SPCs and NPC on the chassis. SRX3400 supports a maximum of up to four SPCs, two NPCs, four IOCs, and six NP-IOCs per chassis. However you must install at least one SPCs and NPC on the chassis. On SRX3400 and SRX3600 devices you must install PICs on the front slots of the chassis. For more details, see [SR X1400](#) , [SRX3400](#) , and [SRX3600 Services Gateway Module Guide](#).

RELATED DOCUMENTATION

[System Memory and Storage Media for SRX Series Firewalls](#) | 35

[Storage Media Names for SRX Series Devices](#)

Upgrade to Junos OS with Upgraded FreeBSD

IN THIS CHAPTER

- [Upgrade and Downgrade for Junos OS with Upgraded FreeBSD | 87](#)

Upgrade and Downgrade for Junos OS with Upgraded FreeBSD

SUMMARY

You can upgrade or downgrade Junos OS with upgraded FreeBSD. You can upgrade Junos OS with upgraded FreeBSD from Junos OS based on FreeBSD 6.1 and upgrade between different releases of Junos OS with upgraded FreeBSD. Before you stage an upgrade between different releases of Junos OS with upgraded FreeBSD, you should install the os-package software to help the upgrade go more smoothly.

IN THIS SECTION

- [Before You Upgrade, Install os-package | 87](#)
- [Install Junos OS with Upgraded FreeBSD Over Junos OS with Upgraded FreeBSD of a Different Release | 89](#)
- [Upgrade to a Junos OS Release with Upgraded FreeBSD | 91](#)
- [Downgrading from Junos OS with Upgraded FreeBSD | 97](#)



NOTE: If you are upgrading or installing Junos OS on a VM host, see ["Installing, Upgrading, Backing Up, and Recovery of VM Host"](#) on page 385.

Before You Upgrade, Install os-package

Before you stage an upgrade between different releases of Junos OS with upgraded FreeBSD, you should install the os-package software to help the upgrade go more smoothly. A vast majority of all upgrade problems are due to limitations in the already running software that is performing the installation, rather than the new software being installed. The os-package software contains the latest version of the package system. The software is installable on any release of Junos OS running an upgraded FreeBSD version (FreeBSD 10 or later).

Benefits:

- os-package facilitates the major FreeBSD upgrades (that is, version 10 to version 11 or version 11 to version 12).
- The goal of the os-package is to be backward compatible with all prior BSDx releases of JUNOS.
- os-package is architecture neutral.

You do not need to reboot the device after installing os-package. It takes only a few seconds to add and is immediately available for help with a planned upgrade. When added, the os-package checks the os-kernel for a feature toggle, which indicates that it is safe to reboot with the os-package in the active set. If the toggle is missing, the following note is issued:

NOTE: os-package will remove itself from 'active' set at next boot.



NOTE: The os-package is NOT bundled with Junos OS Release 22.2R1 and older. os-package is needed only when the shipped JUNOS package that is running on a device needs to be updated to facilitate an upgrade. You should install the latest package before every upgrade regardless of whether the os-package was installed previously on the device. Even when os-package is bundled with Junos OS (Release 22.3R1 and later), you should fetch and install the latest os-package before you upgrade to reduce the likelihood of issues impacting the upgrade.

Before you install os-package:

- Determine which Junos OS releases have BSDx, by platform: [Junos kernel upgrade to FreeBSD 10+](#)
 - Once you know which Junos OS BSDx release you have, find the correct os-package for it:
 - Junos OS Release 18.x and later: [os-package](#) (For example, os-package-20221105.013526_builder_stable_12.tgz.)
 - Junos OS Release 17.x and earlier: [os-package-sha1](#) (For example, os-package-sha1-20221105.013526_builder_stable_12.tgz.)
1. Download the latest copy of os-package for your version of Junos OS. Save the copy to the **/var/tmp** folder of the device.

The original filename of os-package looks similar to this example: os-package-20221105.013526_builder_stable_12.tgz. In this procedure, we are using the filename os-package.tgz to refer to the package.

2. Install os-package.

```
root@juniper> request system software add /var/tmp/os-package.tgz
```



NOTE: It only takes a few seconds to run and does not need a reboot.

Do not add os-package when there is already a 'pending' set, else you will get no benefit from os-package. Thus, if you see the following notice after adding os-package, you'll need to rollback the software:

```
NOTICE: 'pending' set will be activated at next reboot...
```

```
root@juniper> request system software rollback
```

3. If you had to rollback the software in the previous step due to a pending set, you need to repeat steps 1 and 2 to install os-package.

Once os-package has successfully installed, you can proceed to upgrade to a higher version of Junos OS. If the system reboots before you've had a chance to upgrade Junos OS, os-package deactivates itself, and you will have to install os-package again. If you ever want to delete os-package, you can use the `request system software delete os-package` CLI command.

Install Junos OS with Upgraded FreeBSD Over Junos OS with Upgraded FreeBSD of a Different Release



CAUTION: If you do a media install (either USB or network), the system is wiped and re-partitioned completely. Before you begin, if you have important files, copy them from the device to a secure location before upgrading the device.

To install Junos OS with upgraded FreeBSD over Junos OS with upgraded FreeBSD of a different release:

1. Enter the `request system software add package-name validate reboot` command from the operational mode in the CLI:



NOTE: Because Junos OS Release 21.2R1 runs on FreeBSD 12, which uses system calls not available on FreeBSD 10 or 11, you must include one of the following options

instead of the `validate` option on the `request system software add` command when installing the package:

- `no-validate`
- `validate-on-host`
- `validate-on-routing-engine`



NOTE: The `no-copy` option is ignored.

Use the `validate` and `reboot` options with the `request system software add` command. The command uses the `validate` option by default. We encourage users to validate using the `validate` option when upgrading from Junos OS to Junos OS or from Junos OS with upgraded FreeBSD to Junos OS with upgraded FreeBSD.

If you leave out the `reboot` option, you can take care of that in a separate reboot step.

The new Junos OS image is installed on the device.

2. Verify the installation of Junos OS with upgraded FreeBSD.

```
user@host> show version
```



NOTE: The output shows the OS kernel, OS runtime, and other packages installed on the device.

3. If you previously left off the `reboot` option in Step 1, reboot the device to start the new software using the `request system reboot` command:

```
user@host> request system reboot
Reboot the system? [yes, no] (no) yes
```



NOTE: You must reboot the device to load the newly installed version of Junos OS with upgraded FreeBSD on the device.

To terminate the installation, do not reboot the device. Instead, finish the installation and then issue the `request system software rollback install-package-name.tgz` command. This is your last chance to stop the installation (not applicable on EX2300 and EX3400 platforms).

The software is loaded when you reboot the system. Installation can take between 5 and 10 minutes. The device then reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation does not route traffic.

Upgrade to a Junos OS Release with Upgraded FreeBSD

SUMMARY

This topic discusses how to upgrade to a release of Junos OS with upgraded FreeBSD.

IN THIS SECTION

- [Determine Which Package or Packages to Install | 92](#)
- [Install a Junos OS with Upgraded FreeBSD Release Over an Existing Junos OS Release | 94](#)



NOTE: If you are upgrading or installing Junos OS on a VM host, see ["Installing, Upgrading, Backing Up, and Recovery of VM Host"](#) on page 385.

This section covers upgrading from Junos OS based on FreeBSD 6.1 to Junos OS with upgraded FreeBSD. It does not address upgrading using ISSU. There are certain limitations to using ISSU when upgrading to Junos OS with upgraded FreeBSD. For more information on using ISSU, see [Example: Performing a Unified ISSU](#).

When you are upgrading to a different release of Junos OS, you usually use the `request system software add validate` command. The `validate` option checks the candidate software against the current configuration of the device to ensure they are compatible. (Validate is the default behavior when the software package being added is a different release.) However, there are circumstances under which you cannot validate the running configuration in this way. One such circumstance is when you are upgrading to Junos OS with upgraded FreeBSD from Junos OS based on FreeBSD 6.1. Another such circumstance is when you are updating between different releases of Junos OS with upgraded FreeBSD, and the newest version of FreeBSD uses system calls that are not available in earlier versions of FreeBSD.

If you are upgrading between releases that cannot use direct validation, you need to specify one of the following on the `request system software add operational mode` command when you upgrade:

- The `no-validate` option—this option does not validate the software package against the current configuration. Therefore, the current configuration might fail once you upgrade the system. Choose this option for the first time you upgrade to the newer version.

- The `validate-on-host` option—this option validates the software package by comparing it to the running configuration on a remote Junos OS host. Be sure to choose a host that you have already upgraded to the newer version of software.
- The `validate-on-routing-engine` option—(for systems with redundant Routing Engines) this option validates the software package by comparing it to the running configuration on a Routing Engine in the same chassis. Use this option when you have already upgraded the other Routing Engine to the newer version.

If you are upgrading between releases that cannot use direct validation, another approach would be to validate on a different host. It does not matter where that other host is, as long as you can reach it with NETCONF over SSH (see *Establishing an SSH Connection for a NETCONF Session*). The target system uses the network to contact the other host, run the validation and authentication, and return the result.

The procedure covers upgrading to a release of Junos OS with upgraded FreeBSD from a release of Junos OS based on FreeBSD 6.1. To determine whether you are upgrading between releases that can use direct validation or not, see ["Upgrade to a Junos OS Release with Upgraded FreeBSD" on page 91](#).



NOTE: Before installing software on a device that has one or more custom YANG data models added to it, back up and remove the configuration data corresponding to the custom YANG data models from the active configuration. For more information see ["Managing YANG Packages and Configurations During a Software Upgrade or Downgrade" on page 205](#).

Determine Which Package or Packages to Install

To determine which software package to install to upgrade to Junos OS with upgraded FreeBSD, you will need to consult the Feature Explorer and Table 1.

You can skip no more than two releases when upgrading (or downgrading). That means you can upgrade only to one of the three releases subsequent to your current release. If you want to upgrade across more releases than this, you need to perform multiple upgrades.

We recommend you upgrade to a 64-bit image of Junos OS with upgraded FreeBSD. In Junos OS releases earlier than 15.1, the partition swap pages are counted as part of the memory file system partition. Using this method leaves 4 GB of memory as the maximum that is theoretically accessible when you are using a 32-bit image. However, when Junos OS with upgraded FreeBSD is run, the system only counts the actual partition size, which leaves around 3.4 GB of available physical address space, or only 3 GB of usable RAM.

To determine which installation package and procedure you require:

1. See the **Junos kernel upgrade to FreeBSD 10+** entry in [Feature Explorer](#).

Click the link or go to <https://apps.juniper.net/feature-explorer/>, type **freebsd**, and select **Junos kernel upgrade to FreeBSD 10+**.

You will see a listing of platforms that run Junos OS with upgraded FreeBSD and the software release it was introduced in. Different platforms first support Junos OS with upgraded FreeBSD in different releases. Use this listing to find which release you need to install for your device to upgrade to Junos OS with FreeBSD.

2. Consult Table 1 to determine the upgrade path to follow.

- Determine which release your device is currently running.

Look first at the release sequence and then at the second column and find the release running on your device.

- Determine which release you need to install.

The third column will give you the earliest release you need to install for your platform type to be running Junos OS with upgraded FreeBSD.

Table 13: Upgrade Path to Junos OS with the Upgraded FreeBSD

Release Sequence	Current Router's Junos OS Release	Earliest Release Supporting Junos OS with Upgraded FreeBSD	Upgrade Path	Example
Routing and Switching	All currently supported routers and switches run Junos OS with Upgraded FreeBSD.	All currently supported routers and switches run Junos OS with Upgraded FreeBSD.	Upgrade in a single step.	To upgrade from Release 23.4R1, upgrade directly to either Release 24.2R1 or 24.4R1.
Security	12.3 to 17.2	17.3	Upgrade in a single step.	To upgrade from Release 12.3X48, upgrade directly to Release 17.3.
	15.1 to 17.2	17.3	Upgrade in a single step.	To upgrade from Release 15.1X49, upgrade directly to Release 17.3.

Table 13: Upgrade Path to Junos OS with the Upgraded FreeBSD (Continued)

Release Sequence	Current Router's Junos OS Release	Earliest Release Supporting Junos OS with Upgraded FreeBSD	Upgrade Path	Example
	15.1 to 17.3	17.4	Upgrade in a single step.	To upgrade from Release 15.1x49-D80, upgrade directly to Release 17.4.
	(SRX300, SRX320, SRX340, SRX345, SRX380 only), 23.4R2-S3 or 24.2R2	24.4R1	Upgrade in a single step.	For upgrade instructions, including for upgrading from releases earlier than 23.4R2-S3, see <i>KB85650</i> .



NOTE: You can also downgrade from a Junos OS release with upgraded FreeBSD to a Junos OS release based on FreeBSD 6.1 as long as the path complies with the Junos OS policy of skipping at most two earlier releases.

3. Download the Junos OS with upgraded FreeBSD package.

For a table listing the package prefixes, see ["Junos OS and Junos OS Evolved Installation Package Names" on page 15](#).

4. Continue installing a software package on a device by using the procedure in this topic: ["Install a Junos OS with Upgraded FreeBSD Release Over an Existing Junos OS Release" on page 94](#).

Install a Junos OS with Upgraded FreeBSD Release Over an Existing Junos OS Release

Upgrading to a release of Junos OS with upgraded FreeBSD reformats the file system. Unless you take precautions to save important files elsewhere, most files and directories are deleted. By default, the upgrade process preserves only the following directories and the files within them:

- `/config`
- `/etc/localtime`

- `/var/db`
- `/var/etc/master.passwd`
- `/var/etc/inetd.conf`
- `/var/etc/pam.conf`
- `/var/etc/resolv.conf`
- `/var/etc/syslog.conf`
- `/var/etc/localtime`
- `/var/etc/exports`
- `/var/etc/extensions.allow`
- `/var/preserve`
- `/var/tmp/baseline-config.conf`
- `/var/tmp/preinstall_boot_loader.conf`



NOTE: In the `/var/db/config` directory, up to 10 rollback configurations are saved, depending on the configuration file size.



NOTE: On EX2300 and EX3400 switches, the following directories are not applicable:

- `/etc/localtime`
- `/var/etc/localtime`
- `/var/etc/exports`
- `/var/preserve`
- `/var/tmp/preinstall_boot_loader.conf`

Before you begin, if you have important files in directories that are not preserved, copy them from the device to a secure location before upgrading the device.



CAUTION: If you do a media install (either USB or network), the system is wiped and re-partitioned completely. Before you begin, if you have important files, copy them from the device to a secure location before upgrading the device.

To install a release of Junos OS with upgraded FreeBSD over plain Junos OS:

1. Enter the request `system software add install-package-name.tgz no-validate` command from the operational mode in the CLI:



NOTE: The `no-copy` option is enabled by default.

Use the `no-validate` option with the `request system software add` command. If you leave out the `no-validate` option, the command uses the `validate` option by default, and direct validation of the running configuration does not work for upgrading to Junos OS with upgraded FreeBSD from Junos OS based on older versions of the FreeBSD kernel.



NOTE: You can also use the `reboot` option along with the `request system software add` command, but it is not recommended to do this in a single step while upgrading from a FreeBSD 6.1 based Junos OS to Junos OS with upgraded FreeBSD.



NOTE: To validate the current configuration on an upgrade to Junos OS with upgraded FreeBSD from Junos OS, use the `request system software validate on (Junos OS with Upgraded FreeBSD)` command.

```
user@host>request system software add /var/tmp/install-package-name.tgz no-validate
```

The new Junos OS image is installed on the device.

2. Reboot the device to start the new software using the `request system reboot` command:

```
user@host> request system reboot
Reboot the system? [yes, no] (no) yes
```



NOTE: You must reboot the device to load the newly installed version of Junos OS on the device.

To terminate the installation, do not reboot the device. Instead, finish the installation and then issue the `request system software delete install-package-name.tgz` command. This is your last chance to stop the installation (not applicable on EX2300 and EX3400 platforms).

The software is loaded when you reboot the system. Installation can take between 5 and 10 minutes. The device then reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation does not route traffic.

3. Log in and issue the `show version` command to verify the version of the software installed.



NOTE: The output shows the OS kernel, OS runtime, and other packages installed on the device.

RELATED DOCUMENTATION

Establishing an SSH Connection for a NETCONF Session

Downgrading from Junos OS with Upgraded FreeBSD

SUMMARY

This topic discusses the different procedures for downgrading from a release of Junos OS with upgraded FreeBSD.

IN THIS SECTION

- [Downgrading from Junos OS with Upgraded FreeBSD to Legacy Junos OS | 98](#)
- [Downgrading from Junos OS with Upgraded FreeBSD Release 17.4 or Later to Release 15.1 Through 17.3 | 99](#)
- [Downgrading from Junos OS with Upgraded FreeBSD Release 17.3 or Earlier to Release 15.1 Through 17.2 | 100](#)
- [Downgrading from Junos OS with Upgraded FreeBSD Release 18.1 or Later to Release 17.4 or Later | 101](#)

Certain hardware platforms run a Junos OS based on an upgraded FreeBSD kernel instead of older versions of FreeBSD. To find which platforms support Junos OS with upgraded FreeBSD, see [Feature Explorer](#), enter `freebsd`, and select **Junos kernel upgrade to FreeBSD 10+**.



NOTE: If you are upgrading or installing Junos OS on a VM host, see ["Installing, Upgrading, Backing Up, and Recovery of VM Host" on page 385](#).

This topic discusses the different procedures for downgrading from a release of Junos OS with upgraded FreeBSD. One procedure describes how to downgrade to legacy Junos OS. The other procedures describe how to downgrade to an earlier release of Junos OS with upgraded FreeBSD.

The main difference between the procedures is whether to use the `validate` or `no-validate` option with the `request system software add` command. If you downgrade between two versions of legacy Junos OS, `validate` works. Similarly, if you downgrade from Junos OS with upgraded FreeBSD Release 18.1 or later to Release 17.4 or later, `validate` works. However, there is one set of circumstances in which the `no-validate` option must be used when downgrading between Junos OS with upgraded FreeBSD releases, and that is when you downgrade from a Junos OS with upgraded FreeBSD Release 17.4 or later to a release earlier than 17.4, that is, Junos OS releases 15.1 through 17.3.

Select and perform the procedure that matches your set of circumstances.

Downgrading from Junos OS with Upgraded FreeBSD to Legacy Junos OS

If you have previously upgraded to Junos OS with upgraded FreeBSD, you can downgrade to an earlier version of Junos OS (that is, legacy Junos OS) as long as the downgrade conforms to the Junos OS policy of skipping at most two earlier releases.



NOTE: For SRX300, SRX320, SRX340, SRX345, and SRX380 firewalls, you must first downgrade to either Junos OS Release 23.4R2-S3 or Release 24.2R2 before downgrading to any other release. Also, if you have chassis clusters, you cannot use the In-Band Cluster Upgrade (ICU) method for this particular downgrade. You can use either the procedure outlined in *KB85650* or the minimal downtime procedure documented in [KB17947 \(Minimal_Downtime_Upgrade_Branch_Mid PDF file\)](#). You must use the `request system software add package-name no-validate` command to downgrade the software.

This example uses the package `/var/tmp/jinstall-13.3R2.7-domestic-signed.tgz` to install legacy Junos OS on the primary Routing Engine (re0).

To downgrade from Junos OS with upgraded FreeBSD to legacy Junos OS:

1. Enter the `request system software add package-name no-validate reboot` command from the operational mode in the CLI.

Use the `no-validate` and `reboot` options with the `request system software add` command. If you leave out the `no-validate` option, the command uses the `validate` option by default, and direct validation of

running configuration does not work for downgrading to legacy Junos OS from Junos OS with upgraded FreeBSD.

If you leave out the reboot option, you can take care of that in a separate reboot step.

The following example uses the `re0` option:

```
user@host> request system software add /var/tmp/jinstall-13.3R2.7-domestic-signed.tgz re0 no-
validate reboot
THIS IS A SIGNED PACKAGE Saving the config files ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install Rebooting. Please wait ...
shutdown: [pid 11001] Shutdown NOW! *** FINAL System shutdown message
from root@host *** System going down IMMEDIATELY Shutdown NOW! System
shutdown time has arrived\x07\x07 users@host> Connection to
device1.example.com closed by remote host. Connection to
device1.example.com closed. ... user@router> show version
Hostname: host
Model: mx240
Junos: 13.3R2.7
JUNOS Base OS boot [13.3R2.7]
JUNOS Base OS Software Suite [13.3R2.7]
JUNOS Kernel Software Suite [13.3R2.7]
JUNOS Crypto Software Suite [13.3R2.7]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [13.3R2.7]
JUNOS Packet Forwarding Engine Support (MX Common) [13.3R2.7]
JUNOS Online Documentation [13.3R2.7]
JUNOS Services AACL Container package [13.3R2.7]
...
```

2. Verify the downgrade of the software package.

```
user@host> show version
```

The output shows the OS kernel, OS runtime, and other packages installed on the device.

Downgrading from Junos OS with Upgraded FreeBSD Release 17.4 or Later to Release 15.1 Through 17.3

This procedure is applicable when downgrading from Junos OS with Upgraded FreeBSD Release 17.4 or later to an earlier release of Junos OS with Upgraded FreeBSD.



NOTE: If you have important files in other directories, copy them from the router or switch to a secure location before upgrading the router or switch.

To downgrade from Junos OS with upgraded FreeBSD Release 17.4 or later to a Release 15.1 through 17.3:

1. Enter the request `system software add package-name no-validate reboot` command from the operational mode in the CLI:

Use the `no-validate` and `reboot` options with the request `system software add` command. If you leave out the `no-validate` option, the command uses the `validate` option by default, and direct validation of running configuration does not work for downgrading to an earlier release of Junos OS with upgraded FreeBSD from Junos OS with upgraded FreeBSD Release 17.4 or later.

If you leave out the `reboot` option, you can take care of that in a separate reboot step.

The new Junos OS image is installed on the device.

2. Verify the installation of Junos OS with upgraded FreeBSD:

```
user@host> show version
```

The output shows the OS kernel, OS runtime, and other packages installed on the device.

Downgrading from Junos OS with Upgraded FreeBSD Release 17.3 or Earlier to Release 15.1 Through 17.2

This procedure is applicable when downgrading from Junos OS with Upgraded FreeBSD Releases 17.3 through 15.1 to an earlier release of Junos OS with Upgraded FreeBSD.



NOTE: If you have important files in other directories, copy them from the router or switch to a secure location before upgrading the router or switch.

To downgrade from Junos OS with upgraded FreeBSD Release 17.3 or earlier to an earlier release of Junos OS with upgraded FreeBSD:

1. Enter the request `system software add package-name validate reboot` command from the operational mode in the CLI:

Use the `validate` and `reboot` options with the request `system software add` command. The command uses the `validate` option by default. If you leave out the `reboot` option, you can take care of that in a separate reboot step.

The new Junos OS image is installed on the device.

2. Verify the installation of Junos OS with upgraded FreeBSD:

```
user@host> show version
```

The output shows the OS kernel, OS runtime, and other packages installed on the device.

Downgrading from Junos OS with Upgraded FreeBSD Release 18.1 or Later to Release 17.4 or Later

This procedure is applicable when downgrading from Junos OS with Upgraded FreeBSD Releases 18.1 or later to a Junos OS with Upgraded FreeBSD Release 17.4 or later.



NOTE: If you have important files in other directories, copy them from the router or switch to a secure location before upgrading the router or switch.

To downgrade from Junos OS with upgraded FreeBSD Release 18.1 or later to Junos OS with Upgraded FreeBSD Release 17.4 or later:

1. Enter the request `system software add package-name validate reboot` command from the operational mode in the CLI:

Use the `validate` and `reboot` options with the request `system software add` command. The command uses the `validate` option by default. If you leave out the `reboot` option, you can take care of that in a separate reboot step.

The new Junos OS image is installed on the device.

2. Verify the installation of Junos OS with upgraded FreeBSD:

```
user@host> show version
```

The output shows the OS kernel, OS runtime, and other packages installed on the device.

RELATED DOCUMENTATION

request system snapshot (Junos OS with Upgraded FreeBSD)

request system reboot (Junos OS with Upgraded FreeBSD)

CHAPTER 5

Install Software on Routers

IN THIS CHAPTER

- [Installing Software on Routing Devices \(Junos OS\) | 102](#)
- [Installing Software on ACX Series Routers \(Junos OS\) | 113](#)
- [Configuring Root Partitions on ACX Series Routers | 118](#)
- [Installing Software on MX Series Routers Using a USB Flash Drive | 127](#)

Installing Software on Routing Devices (Junos OS)

IN THIS SECTION

- [Installing the Software Package on a Router with a Single Routing Engine \(Junos OS\) | 102](#)
- [Installing the Software Package on a Device with Redundant Routing Engines \(Junos OS\) | 104](#)

Routing devices are delivered with Junos OS preinstalled on them. As new features and software fixes become available, you must upgrade Junos OS to use them. You can install software on single and redundant routing engines.

Installing the Software Package on a Router with a Single Routing Engine (Junos OS)

Before you install a new software release on a device, you should back up the current system.



NOTE: Starting in Junos OS release 20.3R1, ACX710 routers support limited images.

To upgrade the software on a router or switch:

1. Install the new software package using the `request system software add` command:

```
user@host> request system software add /var/tmp/installation-package
```

The variable *installation-package* is the name of the installation package. Specify the absolute path on the local disk. For package name prefixes, see ["Junos OS Installation Package Names" on page 15](#).



NOTE: (Junos OS only) To install multiple software packages at one time, you can use the `request system software add set` command. For more information on this command, see the `set` option in *request system software add (Junos OS)*.



CAUTION: Do not include the `re0 | re1` option when you install a package using the `request system software add` command, if the Routing Engine on which the package is located and the Routing Engine on which you want to install the package are the same. In such cases, the package gets deleted after a successful upgrade.

2. Reboot the device to start the new software:

```
user@host> request system reboot
Reboot the system ? [yes,no] (no) yes
```



NOTE: You must reboot the device to load the new software release on the device. To terminate the installation, do not reboot the device. Instead, finish the installation and then issue the `request system software delete package-name` command. This is your last chance to stop the installation.

The software is loaded when you reboot the system. Installation can take between 5 and 10 minutes. The device then reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation does not route traffic.

3. Log in and verify the release of the software installed:

- To verify release for installation of a Junos OS release, use the `show version` command.

```
user@host> show version
```

SEE ALSO

`request system software add (Junos OS)`
[show version](#)

Installing the Software Package on a Device with Redundant Routing Engines (Junos OS)

IN THIS SECTION

- [Preparing the Device for the Installation \(Junos OS\) | 104](#)
- [Installing Software on the Backup Routing Engine \(Junos OS\) | 106](#)
- [Installing Software on the Remaining Routing Engine \(Junos OS\) | 108](#)
- [Finalizing the Installation \(Junos OS\) | 110](#)

If the device has two Routing Engines, perform a Junos OS installation on each Routing Engine separately to minimize disruption to network operation.

To upgrade redundant Routing Engines, you first install the new Junos OS release on the backup Routing Engine while keeping the currently running software version on the primary Routing Engine. After making sure that the new software version is running correctly on the backup Routing Engine, you switch device control to the backup Routing Engine. Finally, you install the new software on the new backup Routing Engine. For detailed procedures, see the following subsections:

Preparing the Device for the Installation (Junos OS)

Determine if this is the best procedure for upgrading your device:

- To upgrade two different Junos OS releases with no disruption on the control plane and with minimal disruption of traffic, you can use In-Service Software Upgrade, see [Understanding Unified ISSU](#) for routers and switches, and [Upgrading a Chassis Cluster Using In-Service Software Upgrade](#) for SRX Series Firewalls.
- To upgrade the software running on EX Series Ethernet Switches with redundant Routing Engines and all member switches in EX Series Virtual Chassis with a single command, you can use Nonstop Software Upgrade, see *Understanding Nonstop Software Upgrade on EX Series Switches*.
- To upgrade the software package on an EX6200 switch or an EX8200 switch with one installed Routing Engine, see ["Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine \(CLI Procedure\)" on page 146](#).



WARNING: If graceful Routing Engine switchover (GRES) or nonstop active routing (NSR) is enabled when you initiate a software installation, the software does not install properly. Make sure you deactivate GRES (if it is enabled). By default, NSR is disabled. If NSR is enabled, remove the nonstop-routing statement from the [edit routing-options] hierarchy level to disable it.

To ensure GRES and NSR are disabled:

1. Log in to the primary Routing Engine's console.

For more information about logging in to the Routing Engine through the console port, see the specific hardware guide for your device.

2. From the CLI operational prompt, enter configuration mode:

```
{master}
user@host> configure
Entering configuration mode

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

3. Disable nonstop active routing (NSR):

```
{master}[edit]
user@host# delete routing-options nonstop-routing
```

4. Disable nonstop-bridging if it is enabled:

```
{master}[edit]
user@host# delete protocols layer2-control nonstop-bridging
```

5. Disable Routing Engine redundancy if enabled:

```
{master}[edit]
user@host# (delete | deactivate) chassis redundancy graceful-switchover
```

6. Save the configuration change on both Routing Engines:

```
{master}[edit]
user@host# commit synchronize
re0:
configuration check succeeds
re1:
commit complete
re0:
commit complete
```



NOTE: To ensure the most recent configuration changes are committed before the software upgrade, perform this step even if nonstop active routing and graceful Routing Engine switchover were previously disabled.

7. Exit the CLI configuration mode:

```
[edit]
user@host# exit
```

Installing Software on the Backup Routing Engine (Junos OS)

After the device has been prepared, you first install the new Junos OS release on the backup Routing Engine, while keeping the currently running software version on the primary Routing Engine. This enables the primary Routing Engine to continue operations, minimizing disruption to your network.

Before you start this procedure, decide which software package you need and download it to the **/var/tmp** directory of the primary Routing Engine. For information on which packages to use for which upgrades, see ["Junos OS Installation Package Names" on page 15](#).

To install software on the backup Routing Engine:

1. Log in to the console port on the current primary Routing Engine in slot 0.
2. Install the new software package on the backup Routing Engine (re1) using the `request system software add` command:

```
user@host> request system software add re1 validate /var/tmp/jinstall-9.2R1.8-domestic-
signed.tgz
```

Installation and validation take about 15 minutes.



CAUTION: Do not include the `re0` or `re1` option when you install a package using the `request system software add` command if the Routing Engine on which the package is located and the Routing Engine on which you want to install the package are the same. In such cases, the package gets deleted after a successful upgrade.

For MX Series you can use the `request system software add set` command to install multiple software packages at the same time:

```
user@host> request system software add set re1 /var/tmp/installation-package
```

For more information about the `request system software add set` command, see *request system software add (Junos OS)* or the [CLI Explorer](#).

3. Reboot the backup Routing Engine to start the new software:

```
user@host> request system reboot other-routing-engine
Rebooting re1
user@host>
```

You must reboot the device to load the new installation of Junos OS on the device. You can combine steps 2 and 3 by adding **reboot** to the `request system software add` command. But if you do the steps separately, make sure you reboot the Routing Engine you just added system software to.



NOTE: To terminate the installation, do not reboot your device. Instead, finish the installation and then issue the `request system software delete software-package-name` command. This is your last chance to stop the installation.

All the software is loaded when you reboot the device. Installation can take between 5 and 10 minutes. The device then reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation is not routing traffic.

4. Issue the `show version invoke-on other-routing-engine` command to verify the new software is installed.

```

user@host> show version invoke-on other-routing-engine
re1:
-----
Hostname: host1
Model: mx240
Junos: package-name
. . .
user@host>

```

5. (Optional) Add the **jweb** package using the `request system software add` command. Before you can add this package, you must first download the software as you did the installation package. For more information about downloading the **jweb** package, see ["Downloading Software \(Junos OS\)" on page 73](#).

The **jweb** installation module adds a router management graphical user interface that you can use to view and configure your router.

Installing Software on the Remaining Routing Engine (Junos OS)

Once the software is installed on the backup Routing Engine, you are ready to switch routing control to the backup Routing Engine, and then upgrade or downgrade the software on the remaining Routing Engine in slot 0.

To install software on the primary Routing Engine:

1. Transfer routing control from the primary to the backup Routing Engine:

```

user@host> request chassis routing-engine master switch
warning: Traffic will be interrupted while the PFE is re-initialized
Toggle mastership between routing engines ? [yes,no] (no) yes

Resolving mastership...
Complete. The other routing engine becomes the master.

```

For more information about the `request chassis routing-engine master` command, see the [CLI Explorer](#).

2. Verify that the Routing Engine in slot 1 is now the primary Routing Engine:

```

user@host> show chassis routing-engine

```

Routing Engine status:

Slot 0:

Current state	Backup
Election priority	Master (default)

Routing Engine status:

Slot 1:

Current state	Master
Election priority	Backup (default)

3. Install the new software package on the Routing Engine in slot 0 using the `request system software add` command:

```
user@host> request system software add validate re0 /var/tmp/jinstall-9.2R1.8-domestic-
signed.tgz
```

Installation and validation take about 15 minutes.



CAUTION: Do not include the `re0` or `re1` option when you install a package using the `request system software add` command if the Routing Engine on which the package is located and the Routing Engine on which you want to install the package are the same. In such cases, the package gets deleted after a successful upgrade.

For M Series, MX Series, and T Series routers running Junos OS Release 12.2 and later, you can use the `request system software add set` command to install multiple software packages at the same time:

```
user@host> request system software add set re0 /var/tmp/installation-package
```

For more information about the `request system software add set` command, see *request system software add* (Junos OS) or the [CLI Explorer](#).

4. Reboot the Routing Engine using the `request system reboot` command:

```
user@host> request system reboot
Reboot the system? [yes, no] (no) yes
```

You must reboot the device to load the new installation of Junos OS on the device. You can combine steps 3 and 4 by adding **reboot** to the `request system software add` command. But if you do the steps separately, make sure you reboot the Routing Engine you just added system software to.



NOTE: To terminate the installation, do not reboot your device. Instead, finish the installation and then issue the request system software delete *software-package-name* command. This is your last chance to stop the installation.

The software is loaded when you reboot the system. Installation can take between 5 and 10 minutes. The device then reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation does not route traffic.

5. Log in to the current backup Routing Engine (slot 0) and issue the `show version` command to verify the version of the software installed.

```
user@host> show version
```

6. (Optional) Add the **jweb** package using the `request system software add` command. Before you can add this package, you must first download the software as you did the installation package. For more information about downloading the **jweb** package, see ["Downloading Software \(Junos OS\)" on page 73](#).

The **jweb** installation module adds a router management graphical user interface that you can use to view and configure your router.

Finalizing the Installation (Junos OS)

Once the software is installed on both Routing Engines, you return the router back to its original configuration and back up the new installation.

To finalize the redundant Routing Engines upgrade:

1. Restore the configuration that existed before you started this procedure (from Preparing the Device for the Installation (Junos OS)):

```
user@host> configure
[edit]
user@host# rollback 1
```



NOTE: The number on the `rollback` command should match the number of commits you did in preparing the router for the installation. For example, if you did a separate commit for disabling Routing Engine redundancy and disabling nonstop-bridging, you need to use `rollback 2` in this step.

2. Save the configuration change on both Routing Engines:

```
[edit]
user@host# commit synchronize and-quit
```

3. Transfer routing control back to the original primary Routing Engine in slot 0:

```
{backup}
user@host> request chassis routing-engine master switch
warning: Traffic will be interrupted while the PFE is re-initialized
Toggle mastership between routing engines ? [yes,no] (no) yes
Resolving mastership...
Complete. The other routing engine becomes the master.
```

For more information about the `request chassis routing-engine master` command, see the [CLI Explorer](#).

4. Verify that the Routing Engine (slot 0) is indeed the primary Routing Engine:

```
{master}
user@host> show chassis routing-engine

Routing Engine status:
Slot 0:
  Current state           Master
  Election priority       Master (default)

Routing Engine status:
Slot 1:
  Current state           Backup
  Election priority       Backup (default)
```

5. After you have installed the new software and are satisfied that it is successfully running, back up the new software on both the primary and the backup Routing Engines.

- For backing up Junos OS with upgraded FreeBSD, use the `request system snapshot recovery` command. To find which platforms in which releases use Junos OS with upgraded FreeBSD, see [Feature Explorer](#) and enter **Junos kernel upgrade to FreeBSD 10+**.
- For Junos OS, use the `request system snapshot` command:

```
{master}
user@host> request system snapshot
{master}
user@host> request routing-engine login other-routing-engine
{backup}
user@host-re1> request system snapshot
{backup}
user@host-re1> request routing-engine login other-routing-engine
{master}
user@host>
```

The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. The root and **config** file systems are on the router's CompactFlash card, and the **/altroot** and **/altconfig** file systems are on the router's hard disk or solid-state drive (SSD).

For more information about the `request system snapshot` command, see the [CLI Explorer](#).



NOTE: After you issue the `request system snapshot` command, you cannot return to the previous version of the software because the running copy and backup copy of the software are identical.

RELATED DOCUMENTATION

[Understanding Routing Engine Redundancy](#)

[Repartitioning Routing Engine System Storage to Increase the Swap Partition \(Junos OS\) | 28](#)

Installing Software on ACX Series Routers (Junos OS)

IN THIS SECTION

- Installing Junos OS Using a USB Storage Device on ACX Series Routers | 113
- Installing Junos OS Upgrades from a Remote Server on ACX Series Routers | 117

ACX Series routers are delivered with preinstalled Junos operating system (Junos OS). Before you start this procedure, decide which software package you need and download it. For information on which packages to use for which upgrades, see "[Junos OS Installation Package Names](#)" on page 15.

Installing Junos OS Using a USB Storage Device on ACX Series Routers

You can install the Junos OS image on ACX Series routers using a bootable USB storage device.

Before you begin, ensure the following requirements are met:

- For ACX710 USB upgrades, the USB brand is Transcend or Kingston.
- The USB storage device has enough capacity to accommodate the size of the desired Junos OS image.
- The USB storage device is empty and formatted as FAT32.
- On the device running Junos OS, you have physical access to the USB port and console access to the device.

The following sections outline how to download the software image, create a bootable USB storage device, and install the software image on the device running Junos OS.

Download the Software Image

To download the Junos OS install media image to your router:

1. Access the Juniper Networks downloads site at <https://support.juniper.net/support/downloads/>.
2. Select your product.
3. In the drop-down menus, select the OS and version.
4. Expand the **Install Media** section.
5. In the **Downloads** column, click the link for the **USB Install Image** file.

6. Review and accept the End User License Agreement.
7. Follow the instructions on the download page to download the file and save it to the **/var/tmp** directory on the router.

Create a Bootable USB Storage Device

To create a bootable USB using a router running Junos OS:

1. On the router, enter the shell as the root user:

```
user@router> start shell user root
Password:
```

2. Decompress the Junos OS image that you downloaded to the router.

```
root@router:~ # gunzip /path/to/image.img.gz
```

For example:

```
root@router:~ # gunzip /var/tmp/junos-install-media-usb-acx-arm-64-24.2R1.17.img.gz
```

3. On devices where Junos OS runs as a VM over a Linux-based host (VM host), log in to the VM host shell to see the USB activity.

```
root@router:~ # vhcclient -s
root@router-node:~#
```

4. Before inserting the USB device, list the contents of **/dev/**, for example:

```
root@router:~ # ls /dev/da*
/dev/da0 /dev/da0s1c /dev/da0s2a /dev/da0s3 /dev/da0s3e
/dev/da0s1 /dev/da0s1f /dev/da0s2c /dev/da0s3c
/dev/da0s1a /dev/da0s2 /dev/da0s2f /dev/da0s3d

root@router-node:~ # ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sda4 /dev/sda5 /dev/sda6
/dev/sdb /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4 /dev/sdb5 /dev/sdb6
```



NOTE: Your output might differ based on the device you are using. Connect to the device using the console before inserting the USB to see the name given to the USB storage device.

5. Insert the USB storage device into the USB port on the router.
6. Determine the device ID of the USB by using one or more of the following methods.
 - Review the console messages, which should show the device ID when you insert the USB storage device. The messages and device ID can vary based on the platform, for example:

```
da2 at router-sim1 bus 1 target 0 lun 0
da2: <TOSHIBA TransMemory 5.00> Removable Direct Access SCSI-0 device
```

- List the contents of `/dev/` and compare it to the list of device IDs from Step "4" on page 114 to identify the newly added device. The device name can vary by platform, for example:

```
root@router# ls /dev/da*
/dev/da0 /dev/da0s1c /dev/da0s2a /dev/da0s3 /dev/da0s3e
/dev/da0s1 /dev/da0s1f /dev/da0s2c /dev/da0s3c /dev/da2
/dev/da0s1a /dev/da0s2 /dev/da0s2f /dev/da0s3d /dev/da2s1
```

or

```
root@router-node:~# ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sda4 /dev/sda5 /dev/sda6
/dev/sdb /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4 /dev/sdb5 /dev/sdb6
/dev/sdc /dev/sdc1
```

- Issue the `show log messages operational mode` command or the `dmesg` shell command to check the log files for messages related to the USB device name. For example:

```
root@router-node:~# dmesg | tail
...
[ 493.244240] scsi 6:0:0:0: Direct-Access      JetFlash Transcend 16GB   1100 PQ: 0 ANSI: 6
[ 493.244468] sd 6:0:0:0: Attached scsi generic sg2 type 0
[ 493.245149] sd 6:0:0:0: [sdc] 30197760 512-byte logical blocks: (15.5 GB/14.4 GiB)
[ 493.245876] sd 6:0:0:0: [sdc] Write Protect is off
[ 493.245879] sd 6:0:0:0: [sdc] Mode Sense: 43 00 00 00
```

```
[ 493.246653] sd 6:0:0:0: [sdc] Write cache: enabled, read cache: enabled, doesn't
support DPO or FUA
[ 493.251845] sdc: sdc1
[ 493.254346] sd 6:0:0:0: [sdc] Attached SCSI removable disk
```

7. (ACX710 only) Clear the USB storage device by writing zeroes to the device. Ensure that you specify the device ID of the USB storage device as identified in the previous steps. This process might take some time, so do not kill the session.

```
root@router:~# dd if=/dev/zero of=/dev/device-id bs=1m
```

For example, if the device ID is `/dev/da2`, then you would issue the following command:

```
root@router:~# dd if=/dev/zero of=/dev/da2 bs=1m

dd: /dev/da2: end of device
29164+0 records in
29163+0 records out
30579621888 bytes transferred in 2402.249018 secs (12729580 bytes/sec)
```

8. Use the `dd` command to copy the Junos OS image to the USB storage device.

```
root@router:~# dd if=/path/to/image.img of=/dev/device-id bs=1m
```

For example, the following command copies the Junos OS image from the `/var/tmp` directory to the USB with device ID `/dev/da2`.

```
root@router:~# dd if=/var/tmp/junos-install-media-usb-acx-arm-64-24.2R1.17.img of=/dev/da2
bs=1m
```

Install the Junos OS Image

Perform the following steps to install the Junos OS image from the USB storage device:

1. Insert the USB storage device into the USB port.
2. Reboot the router by doing one of the following:
 - Press the power button on the chassis.

- Switch off and turn on the power button behind the Routing Engine.
- Enter the request system reboot command from the CLI.

The system LED starts blinking in green.

On the console, a message is displayed stating that your flash memory device (NAND Flash device) will be formatted and you will lose all the data. You are prompted to confirm the formatting of the flash memory device.

3. Press **y** to confirm and proceed with the formatting process. The flash memory device is formatted and the image is installed on both the partitions.
4. After the installation is completed, the console displays a message prompting you to eject the USB storage device and to press **Enter** to reboot the device.
5. After you remove the USB storage device and press **Enter**, the reboot begins. After the router reboots, the new Junos OS version is loaded and functional. The LED glows steadily in green.



NOTE: If an installation error occurs, the LEDs turn red. You must have console access to the router to troubleshoot an installation error.

Installing Junos OS Upgrades from a Remote Server on ACX Series Routers

You can use the CLI to install Junos OS packages that are downloaded with FTP or HTTP from the specified location on internal media, such as the NAND Flash device.

Before you begin:

- Verify the available space on the NAND Flash device.
- Download the Junos OS package.

To install Junos OS upgrades from a remote server, enter the following command from operational mode:

```
user@host> request system software add junos-juniper-12.2R1.9-domestic.tgz no-copy no-validate
reboot
```

The new Junos OS image is installed on the router and the device is rebooted.



NOTE: On ACX5048 and ACX5096 routers, use the force-host option to force installing the latest version of the Host OS.

```
user@host> request system software jinstall-acx5k-15.1X54-D20.6-domestic-signed.tgz
force-host add validate reboot
```

RELATED DOCUMENTATION

[Configuring Root Partitions on ACX Series Routers](#) | 118

Configuring Root Partitions on ACX Series Routers

IN THIS SECTION

- [Dual-Root Partitioning ACX Series Routers Overview](#) | 118
- [Understanding How the Primary Junos OS Image with Dual-Root Partitioning Recovers on the ACX Series Router](#) | 120
- [Junos OS Release 12.2 or Later Upgrades with Dual-Root Partitioning on ACX Series Routers](#) | 121
- [Example: Installing Junos OS and Configuring a Dual-Root Partition on ACX Series Routers Using the CLI](#) | 122

The dual-root partitions help your ACX Series routers to remain functional even if the file system is corrupted. Also, it helps to recover the file system in case of corruption.

Dual-Root Partitioning ACX Series Routers Overview

IN THIS SECTION

- [Boot Media and Boot Partition on the ACX Series Routers](#) | 119
- [Important Features of the Dual-Root Partitioning Scheme](#) | 119

Dual-root partitioning allows the ACX Series router to remain functional even if there is file system corruption and to facilitate easy recovery of the file system. Dual-root partitioning means that the

primary and backup Junos OS images are kept in two independently bootable root partitions. If the primary root partition becomes corrupted, the system can still boot from the backup Junos OS image located in the other root partition and remain fully functional.



NOTE: ACX5048 and ACX5096 routers do not support dual-root partitioning. All other ACX routers run with dual-root partitioning.

Boot Media and Boot Partition on the ACX Series Routers

With dual-root partitioning, the ACX Series router first tries to boot the Junos OS from the primary root partition and then from the backup root partition on the internal NAND flash. If both primary and backup root partitions of the internal NAND flash fail to boot, you must insert a USB storage media with a copy of the Junos OS from which to boot.

The following is the storage media available on the ACX Series router:

- USB media emergency boot device



NOTE: The USB media device is not dual-root partitioned.

- Dual, internal NAND flash device (first daOs1, then daOs2)

Important Features of the Dual-Root Partitioning Scheme

The dual-root partitioning scheme has the following important features:

- The primary and backup copies of Junos OS images reside in separate partitions. The partition containing the backup copy is mounted only when required. With the single-root partitioning scheme, there is one root partition that contains both the primary and the backup Junos OS images.
- The `request system software add` command for a Junos OS package erases the contents of the other root partition. The contents of the other root partition will not be valid unless software installation is completed successfully.
- Add-on packages, such as `jais` or `jfirmware`, can be reinstalled as required after a new Junos OS image is installed.
- The `request system software rollback` command does not delete the current Junos OS image. It is possible to switch back to the image by issuing the `rollback` command again.

Understanding How the Primary Junos OS Image with Dual-Root Partitioning Recovers on the ACX Series Router

If the ACX Series Universal Metro router is unable to boot from the primary Junos OS image and boots up from the backup Junos OS image in the backup root partition, a message appears on the console at the time of login indicating that the device has booted from the backup Junos OS image.



NOTE: ACX5048 and ACX5096 routers do not support dual-root partitioning.

```
login: user
```

```
Password:
```

```
*****
```

```
**                                                                 **
```

```
**  WARNING: THIS DEVICE HAS BOOTED FROM THE BACKUP JUNOS IMAGE  **
```

```
**                                                                 **
```

```
**  It is possible that the active copy of JUNOS failed to boot up  **
```

```
**  properly, and so this device has booted from the backup copy.  **
```

```
**                                                                 **
```

```
**  Please re-install JUNOS to recover the active copy in case    **
```

```
**  it has been corrupted.                                         **
```

```
**                                                                 **
```

```
*****
```

Because the system is left with only one functional root partition, you should immediately restore the primary Junos OS image using one of the following methods:

- Install a new image using the CLI. When you install the new image, the new image is installed on only one partition—the alternate partition, meaning the router is now running two images. When you reboot, the router boots from the newly installed image, which becomes the primary image. So now there are two different images running on the router. Run the installation process again to update the other partition.
- Use a snapshot of the backup root partition by entering the `request system snapshot slice alternate` command. After the primary root partition is recovered using this method, the device will successfully boot from the primary root partition on the next reboot. After the procedure, the primary root partition will contain the same version of Junos OS as the backup root partition.



NOTE: You can use the CLI command `request system snapshot slice alternate` to back up the currently running root file system (primary or secondary) to the other root partition on the system.

You can use this command to:

- Save an image of the primary root partition in the backup root partition when the system boots from the primary root partition.
- Save an image of the backup root partition in the primary root partition when the system boots from the backup root partition.



WARNING: The process of restoring the alternate root by using the CLI command `request system snapshot slice alternate` takes several minutes to complete. If you terminate the operation before completion, the alternate root might not have all required contents to function properly.

Junos OS Release 12.2 or Later Upgrades with Dual-Root Partitioning on ACX Series Routers



NOTE: If you are upgrading to Junos OS Release 12.2 without transitioning to dual-root partitioning, use the conventional CLI installation method.

To format the media with dual-root partitioning while upgrading to Junos OS Release 12.2 or later, use either of the following installation methods:



NOTE: ACX5048 and ACX5096 routers do not support dual-root partitioning. All other ACX routers run with dual-root partitioning.

- Installation using a USB storage device. We recommend this method if console access to the system is available and the system can be physically accessed to plug in a USB storage device. See *Installing Junos OS Using a USB Storage Device on ACX Series Routers*.
- Installation from the CLI. We recommend this method only if console access is not available. This installation can be performed remotely. See *Installing Junos OS Upgrades from a Remote Server on ACX Series Routers*.

Example: Installing Junos OS and Configuring a Dual-Root Partition on ACX Series Routers Using the CLI

IN THIS SECTION

- [Requirements | 122](#)
- [Overview | 122](#)
- [Configuration | 123](#)
- [Verification | 126](#)

This example shows how to install Junos OS Release 12.2 or later and configure a dual-root partition on ACX Series routers with the CLI.

Requirements

This example requires an ACX Series router. Before you begin, back up any important data.

Overview

IN THIS SECTION

- [Topology | 123](#)

This example formats the NAND Flash device and installs the new Junos OS image on the media with dual-root partitioning. Install the Junos OS Release 12.2 or later image from the CLI by using the `request system software add` command. Partitions are automatically created on ACX Series routers and no option needs to be manually entered for creating partitions. This command copies the image to the device, and then reboots the device for installation. The device boots with the Release 12.2 or later image installed

with the dual-root partitioning scheme. The formatting and installation process is scheduled to run on the next reboot. Therefore, we recommend that this option be used together with the `reboot` option.



NOTE: The process might take 15 to 20 minutes. The system is not accessible over the network during this time.



CAUTION: Using the `request system software add` command erases the existing contents of the media. Only the current configuration is preserved. You should back up any important data before starting the process.



NOTE: Dual, internal NAND Flash device (first `daOs1`, then `daOs2`) and USB storage device are the storage media available on the ACX Series router. The USB storage device is not dual-root partitioned.

In this example, add the software package `junos-juniper-12.2R1.9-domestic.tgz` with the following options:

- `no-copy` option to install the software package. However, do not save the copies of the package files. You should include this option if you do not have enough space on the internal media to perform an upgrade that keeps a copy of the package on the device.
- `no-validate` option to bypass the compatibility check with the current configuration before installation starts.
- `reboot` option to reboot the device after installation is completed.

Topology

Configuration

IN THIS SECTION

- [Procedure | 124](#)

Procedure

CLI Quick Configuration

To install Junos OS Release 12.2 or later and configure dual-root partitioning on ACX Series routers, copy the following command, paste it in a text file, remove any line break, and then copy and paste the command into the CLI.

From operational mode, enter:

```
user@host> request system software add junos-juniper-12.2R1.9-domestic.tgz no-copy no-validate
reboot
```

Step-by-Step Procedure

To install Junos OS Release 12.2 or later and configure a dual-root partition:

1. Upgrade the ACX Series router to Junos OS Release 12.2 or later using the CLI.
2. Install Junos OS Release 12.2 or later and configure the dual-root partition.

```
user@host> request system software add junos-juniper-12.2R1.9-domestic.tgz no-copy no-
validate reboot
Copying package junos-juniper-12.2R1.9-domestic.tgz to var/tmp/install
Rebooting ...
```

Results

In operational mode, confirm your configuration by entering the `show system storage` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Sample output on a system with dual-root partitioning that displays information about the root partition that is mounted (only one root partition is mounted at a point in time):

```
user@host> show system storage
```

Filesystem	Size	Used	Avail	Capacity	Mounted on
/dev/da0s1a	872M	150M	713M	17%	/

devfs	1.0K	1.0K	0B	100%	/dev
/dev/md0	41M	41M	0B	100%	/packages/mnt/jbase
/dev/md1	183M	183M	0B	100%	/packages/mnt/jkernel-
ppc-12.2I20121026_1217_sranjan					
/dev/md2	30M	30M	0B	100%	/packages/mnt/jpfe-
ACX-12.2I20121026_1217_sranjan					
/dev/md3	9.1M	9.1M	0B	100%	/packages/mnt/
jdocs-12.2I20121026_1217_sranjan					
/dev/md4	55M	55M	0B	100%	/packages/mnt/jroute-
ppc-12.2I20121026_1217_sranjan					
/dev/md5	12M	12M	0B	100%	/packages/mnt/jcrypto-
ppc-12.2I20121026_1217_sranjan					
/dev/md6	1.0G	8.0K	951M	0%	/tmp
/dev/md7	1.0G	448K	950M	0%	/mfs
/dev/da0s1e	92M	18K	91M	0%	/config
procfs	4.0K	4.0K	0B	100%	/proc
/dev/da0s3f	3.9G	3.6G	30M	99%	/var
/dev/da0s3d	447M	2.8M	409M	1%	/var/log

If you are done configuring the device, enter `commit` in configuration mode.

You can issue the `fdisk` command from the Junos prompt to display information about the entire partition format on the NAND Flash device. All ACX Series routers run with dual-root partitioning. The following example displays the partition details on an ACX Series router with dual-root partitions:

```
user@host% fdisk
```

```
***** Working on device /dev/da0 *****
parameters extracted from in-core disklabel are:
cylinders=487 heads=255 sectors/track=63 (16065 blks/cyl)

parameters to be used for BIOS calculations are:
cylinders=487 heads=255 sectors/track=63 (16065 blks/cyl)

Media sector size is 512
Warning: BIOS sector numbering starts with sector 1
Information from DOS bootblock is:
The data for partition 1 is:
sysid 165 (0xa5),(FreeBSD/NetBSD/386BSD)
  start 567, size 1011528 (493 Meg), flag 80 (active)
    beg: cyl 0/ head 9/ sector 1;
```

```

        end: cyl 62/ head 254/ sector 63
The data for partition 2 is:
sysid 165 (0xa5),(FreeBSD/NetBSD/386BSD)
    start 1012662, size 1011528 (493 Meg), flag 0
        beg: cyl 63/ head 9/ sector 1;
        end: cyl 125/ head 254/ sector 63
The data for partition 3 is:
sysid 165 (0xa5),(FreeBSD/NetBSD/386BSD)
    start 2024757, size 3581928 (1748 Meg), flag 0
        beg: cyl 126/ head 9/ sector 1;
        end: cyl 348/ head 254/ sector 63
The data for partition 4 is:
sysid 165 (0xa5),(FreeBSD/NetBSD/386BSD)
    start 5607252, size 2200338 (1074 Meg), flag 0
        beg: cyl 349/ head 9/ sector 1;
        end: cyl 485/ head 254/ sector 63

```

In the preceding example, partition 1 and 2 contain two partitions each internally, a root partition and a configuration partition.

Verification

IN THIS SECTION

- [Verifying the Partitioning Scheme Details | 126](#)

Confirm that the configuration is working properly.

Verifying the Partitioning Scheme Details

Purpose

Verify that the partitioning scheme details on the ACX Series router were configured.

Action

In operational mode, enter the `show system storage` command. For details about the output of this command and the descriptions of the output fields, see [show system storage](#).

RELATED DOCUMENTATION

| [Installing Software on ACX Series Routers \(Junos OS\) | 113](#)

Installing Software on MX Series Routers Using a USB Flash Drive

IN THIS SECTION

- [Pre-Installing Junos OS on a USB Flash Drive | 128](#)
- [Installing Junos OS from a USB Flash Drive | 128](#)
- [Upgrading Junos OS using a USB Flash Drive | 129](#)
- [Installing Software on Routing Devices \(Junos OS\) | 132](#)

You can install or upgrade Junos OS on an MX Series router using a USB flash drive.



NOTE: This topic describes overall MX Series router information.

For details about specific MX Series routers, see the Hardware Guide for that device.

To access the Hardware Guide for a specific device, review [Junos OS by Product](#) and select the device. A link to the Hardware Guide for the device is displayed, along with links to other device specific guides.

Before you begin, determine which software package you need and download it onto the USB flash drive. For information on which packages to use for which upgrades, see ["Junos OS and Junos OS Evolved Installation Package Names" on page 15](#).



NOTE: This topic describes a standard Junos OS installation and upgrade with a Junos OS installation package.

For information about a Junos OS recovery installation, see ["Recovery of Junos OS" on page 328](#).

For information about upgrading to Junos OS with upgraded FreeBSD, see ["Upgrade and Downgrade for Junos OS with Upgraded FreeBSD" on page 87](#).

Pre-Installing Junos OS on a USB Flash Drive

A USB flash drive must have Junos OS stored on it before using the software to install on or upgrade the router.

To pre-install the software on a USB flash drive, ensure the following:

- The USB flash drive meets the MX Series router USB port specifications.
- The USB flash drive is empty and formatted as FAT-32.
- The USB flash drive capacity is large enough to accommodate the size of the desired Junos OS package.)
- A computer to download the software package from the download site and copy it to the USB flash drive.

To download a Junos OS package onto a USB flash drive:

1. Insert the USB flash drive into your computer.
2. Navigate to the download site to download the desired Junos OS package to the USB flash drive.
Recommended download site: [MX Series Software Downloads](#).
3. Choose your router model and version and select a software package to download from the **Install Media** menu.
4. Download the software package.
(Optional) Rename the software package for identification purposes.
5. Eject the USB flash drive when the download has completed.

Installing Junos OS from a USB Flash Drive



NOTE: As a best practice, save a system snapshot of the installed Junos OS image and configuration files for backup purposes before starting the installation procedure.
For details about saving a system snapshot, see *request system snapshot (Junos OS with FreeBSD Prior to Release 10)*.

Use the following steps to install a Junos OS package from a USB flash drive.

1. Ensure that the router has been powered off.
2. Insert the USB flash drive into the USB port on the router.
3. Power on the router.
Powering on the router starts the loader script and checks for a Junos OS package on the USB flash drive.
4. When the install prompt appears, enter **Yes**.

5. When the installation has completed, reboot the router:

```
user@host> request system reboot
```

6. After the reboot has completed, log in and verify that the new version of the software has been properly installed.

```
user@host> show version
```

Upgrading Junos OS using a USB Flash Drive

The procedure to upgrade Junos OS on an MX Series router using a USB flash drive is different than installing Junos OS from a USB flash drive. The install procedure includes booting the device from a USB flash drive and installing a fresh image of Junos OS without preserving configuration files or data files. The upgrade procedure includes replacing the existing Junos OS with a different version of Junos OS on the device while preserving existing configuration files and data files.

The procedure to upgrade includes:

- Copying the Junos OS package from the USB flash drive to the internal storage of the router.
- Upgrading Junos OS on the router.



NOTE: As a best practice, free up storage space on the device and save a system snapshot of the existing Junos OS image and configuration files for backup purposes before starting the upgrade procedure.

For details to free storage space, see *request system storage cleanup (Junos OS)*.

For details about saving a system snapshot, see *request system snapshot (Junos OS with FreeBSD Prior to Release 10)*.

1. On the router, enter the shell as the root user:

```
user@router#> start shell user root
```

Password:

2. List the existing devices on the router.

```
root@router# ls directory name
```

For example:

```
root@router# ls /dev/da*
/dev/da0 /dev/da0s1c /dev/da0s2a /dev/da0s3 /dev/da0s3e
/dev/da0s1 /dev/da0s1f /dev/da0s2c /dev/da0s3c
/dev/da0s1a /dev/da0s2 /dev/da0s2f /dev/da0s3d
```

3. Insert the USB flash drive into the USB port on the router.

The console messages describe the device ID of the USB flash drive.

You can use the results of the **ls *directory name*** command to determine the device ID of the USB flash drive by comparing it to the list of device IDs of the previous step.



NOTE: If the console session is not available while inserting the USB, check the `/var/log` directory for messages related to `da`. For example, use the **show log messages | match da** command to display the messages.

In this example, `/dev/da2s1` is the device ID of the USB flash drive.

```
root@router# router1: TOSHIBA TransMemory, rev 2.00/1.00, addr 3
da2 at router-sim1 bus 1 target 0 lun 0
da2: <TOSHIBA TransMemory 5.00> Removable Direct Access SCSI-0 device
da2: 40.000MB/s transfers
da2: 983MB (2013184 512 byte sectors: 64H 32S/T 983C)

root@router# ls /dev/da*
/dev/da0 /dev/da0s1c /dev/da0s2a /dev/da0s3 /dev/da0s3e
/dev/da0s1 /dev/da0s1f /dev/da0s2c /dev/da0s3c /dev/da2
/dev/da0s1a /dev/da0s2 /dev/da0s2f /dev/da0s3d /dev/da2s1
```

4. Create a directory for the USB drive to mount to.

```
mkdir /var/tmp/directory name
```

For example:

```
root@router# mkdir /var/tmp/usb
```

5. Mount the USB drive to the new directory and check the contents of the directory.

```
mount_msdosfs /dev/usb device ID /var/tmp/ directory name
ls /var/tmp/ directory name
```

For example:

```
root@router# mount_msdosfs /dev/da2s1 /var/tmp/usb
root@router# ls /var/tmp/usb
MX_image.tgz
```

6. Copy the Junos OS package from the USB flash drive to the **/var/tmp/** directory of the router's internal storage.

```
cp /var/tmp/directory name/installation-package-name /var/tmp
```



NOTE: Use the **ls *directory name*** command to verify the Junos OS package was copied to the router's internal storage.

In the example, the MX_image.tgz file on the USB flash drive is copied to the /var/tmp directory on the router.

```
root@router# cp /var/tmp/usb/MX_image.tgz /var/tmp
root@router# ls /var/tmp
MX_image.tgz
```

7. Unmount the USB flash drive after the Junos OS package has been copied to the router's internal storage.

```
umount /var/tmp/directory name
```

For example:

```
root@router# umount /var/tmp/usb
root@% router1: at uhub0 port 1 (addr 3) disconnected
(da1:router-sim1:1:0:0): lost device
```

```
(da1:router-sim1:1:0:0): removing device entry
router1: detached
```

Proceed to the next section to continue upgrading Junos OS.



NOTE: For devices with Routing Engines with VMhost support, you can use the USB device as an emergency boot device. For more information, see ["Creating an Emergency Boot Device for Routing Engines with VM Host Support" on page 416](#). After the `junos-vmhost-install-usb` image is written to the USB drive using the `dd` command, you can boot to the USB using the request `vmhost reboot usb` command.

Installing Software on Routing Devices (Junos OS)

IN THIS SECTION

- [Installing the Software Package on a Router with a Single Routing Engine \(Junos OS\) | 132](#)
- [Installing the Software Package on a Device with Redundant Routing Engines \(Junos OS\) | 134](#)

Routing devices are delivered with Junos OS preinstalled on them. As new features and software fixes become available, you must upgrade Junos OS to use them. You can install software on single and redundant routing engines.

Installing the Software Package on a Router with a Single Routing Engine (Junos OS)

Before you install a new software release on a device, you should back up the current system.



NOTE: Starting in Junos OS release 20.3R1, ACX710 routers support limited images.

To upgrade the software on a router or switch:

1. Install the new software package using the request `system software add` command:

```
user@host> request system software add /var/tmp/installation-package
```

The variable *installation-package* is the name of the installation package. Specify the absolute path on the local disk. For package name prefixes, see ["Junos OS Installation Package Names" on page 15](#).



NOTE: (Junos OS only) To install multiple software packages at one time, you can use the `request system software add set` command. For more information on this command, see the `set` option in *request system software add (Junos OS)*.



CAUTION: Do not include the `re0 | re1` option when you install a package using the `request system software add` command, if the Routing Engine on which the package is located and the Routing Engine on which you want to install the package are the same. In such cases, the package gets deleted after a successful upgrade.

2. Reboot the device to start the new software:

```
user@host> request system reboot
Reboot the system ? [yes,no] (no) yes
```



NOTE: You must reboot the device to load the new software release on the device. To terminate the installation, do not reboot the device. Instead, finish the installation and then issue the `request system software delete package-name` command. This is your last chance to stop the installation.

The software is loaded when you reboot the system. Installation can take between 5 and 10 minutes. The device then reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation does not route traffic.

3. Log in and verify the release of the software installed:

- To verify release for installation of a Junos OS release, use the `show version` command.

```
user@host> show version
```

SEE ALSO

request system software add (Junos OS)

[show version](#)

Installing the Software Package on a Device with Redundant Routing Engines (Junos OS)

IN THIS SECTION

- [Preparing the Device for the Installation \(Junos OS\) | 134](#)
- [Installing Software on the Backup Routing Engine \(Junos OS\) | 136](#)
- [Installing Software on the Remaining Routing Engine \(Junos OS\) | 138](#)
- [Finalizing the Installation \(Junos OS\) | 140](#)

If the device has two Routing Engines, perform a Junos OS installation on each Routing Engine separately to minimize disruption to network operation.

To upgrade redundant Routing Engines, you first install the new Junos OS release on the backup Routing Engine while keeping the currently running software version on the primary Routing Engine. After making sure that the new software version is running correctly on the backup Routing Engine, you switch device control to the backup Routing Engine. Finally, you install the new software on the new backup Routing Engine. For detailed procedures, see the following subsections:

Preparing the Device for the Installation (Junos OS)

Determine if this is the best procedure for upgrading your device:

- To upgrade two different Junos OS releases with no disruption on the control plane and with minimal disruption of traffic, you can use In-Service Software Upgrade, see [Understanding Unified ISSU](#) for routers and switches, and [Upgrading a Chassis Cluster Using In-Service Software Upgrade](#) for SRX Series Firewalls.
- To upgrade the software running on EX Series Ethernet Switches with redundant Routing Engines and all member switches in EX Series Virtual Chassis with a single command, you can use Nonstop Software Upgrade, see *Understanding Nonstop Software Upgrade on EX Series Switches*.
- To upgrade the software package on an EX6200 switch or an EX8200 switch with one installed Routing Engine, see ["Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine \(CLI Procedure\)" on page 146](#).



WARNING: If graceful Routing Engine switchover (GRES) or nonstop active routing (NSR) is enabled when you initiate a software installation, the software does not install properly. Make sure you deactivate GRES (if it is enabled). By default, NSR is disabled. If NSR is enabled, remove the nonstop-routing statement from the [edit routing-options] hierarchy level to disable it.

To ensure GRES and NSR are disabled:

1. Log in to the primary Routing Engine's console.

For more information about logging in to the Routing Engine through the console port, see the specific hardware guide for your device.

2. From the CLI operational prompt, enter configuration mode:

```
{master}
user@host> configure
Entering configuration mode

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

3. Disable nonstop active routing (NSR):

```
{master}[edit]
user@host# delete routing-options nonstop-routing
```

4. Disable nonstop-bridging if it is enabled:

```
{master}[edit]
user@host# delete protocols layer2-control nonstop-bridging
```

5. Disable Routing Engine redundancy if enabled:

```
{master}[edit]
user@host# (delete | deactivate) chassis redundancy graceful-switchover
```

6. Save the configuration change on both Routing Engines:

```
{master}[edit]
user@host# commit synchronize
re0:
configuration check succeeds
re1:
commit complete
```

```
re0:
commit complete
```



NOTE: To ensure the most recent configuration changes are committed before the software upgrade, perform this step even if nonstop active routing and graceful Routing Engine switchover were previously disabled.

7. Exit the CLI configuration mode:

```
[edit]
user@host# exit
```

Installing Software on the Backup Routing Engine (Junos OS)

After the device has been prepared, you first install the new Junos OS release on the backup Routing Engine, while keeping the currently running software version on the primary Routing Engine. This enables the primary Routing Engine to continue operations, minimizing disruption to your network.

Before you start this procedure, decide which software package you need and download it to the `/var/tmp` directory of the primary Routing Engine. For information on which packages to use for which upgrades, see ["Junos OS Installation Package Names" on page 15](#).

To install software on the backup Routing Engine:

1. Log in to the console port on the current primary Routing Engine in slot 0.
2. Install the new software package on the backup Routing Engine (re1) using the `request system software add` command:

```
user@host> request system software add re1 validate /var/tmp/jinstall-9.2R1.8-domestic-
signed.tgz
```

Installation and validation take about 15 minutes.



CAUTION: Do not include the `re0` or `re1` option when you install a package using the `request system software add` command if the Routing Engine on which the package is located and the Routing Engine on which you want to install the package are the same. In such cases, the package gets deleted after a successful upgrade.

For MX Series you can use the `request system software add set` command to install multiple software packages at the same time:

```
user@host> request system software add set re1 /var/tmp/installation-package
```

For more information about the `request system software add set` command, see *request system software add (Junos OS)* or the [CLI Explorer](#).

3. Reboot the backup Routing Engine to start the new software:

```
user@host> request system reboot other-routing-engine
Rebooting re1
user@host>
```

You must reboot the device to load the new installation of Junos OS on the device. You can combine steps 2 and 3 by adding **reboot** to the `request system software add` command. But if you do the steps separately, make sure you reboot the Routing Engine you just added system software to.



NOTE: To terminate the installation, do not reboot your device. Instead, finish the installation and then issue the `request system software delete software-package-name` command. This is your last chance to stop the installation.

All the software is loaded when you reboot the device. Installation can take between 5 and 10 minutes. The device then reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation is not routing traffic.

4. Issue the `show version invoke-on other-routing-engine` command to verify the new software is installed.

```
user@host> show version invoke-on other-routing-engine
re1:
-----
Hostname: host1
Model: mx240
Junos: package-name
. . .
user@host>
```

5. (Optional) Add the **jweb** package using the `request system software add` command. Before you can add this package, you must first download the software as you did the installation package. For more information about downloading the **jweb** package, see ["Downloading Software \(Junos OS\)" on page 73](#).

The **jweb** installation module adds a router management graphical user interface that you can use to view and configure your router.

Installing Software on the Remaining Routing Engine (Junos OS)

Once the software is installed on the backup Routing Engine, you are ready to switch routing control to the backup Routing Engine, and then upgrade or downgrade the software on the remaining Routing Engine in slot 0.

To install software on the primary Routing Engine:

1. Transfer routing control from the primary to the backup Routing Engine:

```
user@host> request chassis routing-engine master switch
warning: Traffic will be interrupted while the PFE is re-initialized
Toggle mastership between routing engines ? [yes,no] (no) yes

Resolving mastership...
Complete. The other routing engine becomes the master.
```

For more information about the `request chassis routing-engine master` command, see the [CLI Explorer](#).

2. Verify that the Routing Engine in slot 1 is now the primary Routing Engine:

```
user@host> show chassis routing-engine

Routing Engine status:
Slot 0:
  Current state      Backup
  Election priority  Master (default)

Routing Engine status:
Slot 1:
  Current state      Master
  Election priority  Backup (default)
```

3. Install the new software package on the Routing Engine in slot 0 using the `request system software add` command:

```
user@host> request system software add validate re0 /var/tmp/jinstall-9.2R1.8-domestic-
signed.tgz
```

Installation and validation take about 15 minutes.



CAUTION: Do not include the `re0` or `re1` option when you install a package using the `request system software add` command if the Routing Engine on which the package is located and the Routing Engine on which you want to install the package are the same. In such cases, the package gets deleted after a successful upgrade.

For M Series, MX Series, and T Series routers running Junos OS Release 12.2 and later, you can use the `request system software add set` command to install multiple software packages at the same time:

```
user@host> request system software add set re0 /var/tmp/installation-package
```

For more information about the `request system software add set` command, see *request system software add (Junos OS)* or the [CLI Explorer](#).

4. Reboot the Routing Engine using the `request system reboot` command:

```
user@host> request system reboot
Reboot the system? [yes, no] (no) yes
```

You must reboot the device to load the new installation of Junos OS on the device. You can combine steps 3 and 4 by adding **reboot** to the `request system software add` command. But if you do the steps separately, make sure you reboot the Routing Engine you just added system software to.



NOTE: To terminate the installation, do not reboot your device. Instead, finish the installation and then issue the `request system software delete software-package-name` command. This is your last chance to stop the installation.

The software is loaded when you reboot the system. Installation can take between 5 and 10 minutes. The device then reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation does not route traffic.

5. Log in to the current backup Routing Engine (slot 0) and issue the `show version` command to verify the version of the software installed.

```
user@host> show version
```

6. (Optional) Add the **jweb** package using the `request system software add` command. Before you can add this package, you must first download the software as you did the installation package. For more information about downloading the **jweb** package, see ["Downloading Software \(Junos OS\)" on page 73](#).

The **jweb** installation module adds a router management graphical user interface that you can use to view and configure your router.

Finalizing the Installation (Junos OS)

Once the software is installed on both Routing Engines, you return the router back to its original configuration and back up the new installation.

To finalize the redundant Routing Engines upgrade:

1. Restore the configuration that existed before you started this procedure (from Preparing the Device for the Installation (Junos OS)):

```
user@host> configure
[edit]
user@host# rollback 1
```



NOTE: The number on the `rollback` command should match the number of commits you did in preparing the router for the installation. For example, if you did a separate commit for disabling Routing Engine redundancy and disabling nonstop-bridging, you need to use `rollback 2` in this step.

2. Save the configuration change on both Routing Engines:

```
[edit]
user@host# commit synchronize and-quit
```

3. Transfer routing control back to the original primary Routing Engine in slot 0:

```
{backup}
user@host> request chassis routing-engine master switch
warning: Traffic will be interrupted while the PFE is re-initialized
Toggle mastership between routing engines ? [yes,no] (no) yes
Resolving mastership...
Complete. The other routing engine becomes the master.
```

For more information about the `request chassis routing-engine master` command, see the [CLI Explorer](#).

4. Verify that the Routing Engine (slot 0) is indeed the primary Routing Engine:

```
{master}
user@host> show chassis routing-engine

Routing Engine status:
  Slot 0:
    Current state           Master
    Election priority       Master (default)

Routing Engine status:
  Slot 1:
    Current state           Backup
    Election priority       Backup (default)
```

5. After you have installed the new software and are satisfied that it is successfully running, back up the new software on both the primary and the backup Routing Engines.

- For backing up Junos OS with upgraded FreeBSD, use the `request system snapshot recovery` command. To find which platforms in which releases use Junos OS with upgraded FreeBSD, see [Feature Explorer](#) and enter **Junos kernel upgrade to FreeBSD 10+**.
- For Junos OS, use the `request system snapshot` command:

```
{master}
user@host> request system snapshot
{master}
user@host> request routing-engine login other-routing-engine
{backup}
user@host-re1> request system snapshot
```

```
{backup}
user@host-re1> request routing-engine login other-routing-engine
{master}
user@host>
```

The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. The root and **/config** file systems are on the router's CompactFlash card, and the **/altroot** and **/altconfig** file systems are on the router's hard disk or solid-state drive (SSD).

For more information about the `request system snapshot` command, see the [CLI Explorer](#).



NOTE: After you issue the `request system snapshot` command, you cannot return to the previous version of the software because the running copy and backup copy of the software are identical.

RELATED DOCUMENTATION

[Understanding Routing Engine Redundancy](#)

[Repartitioning Routing Engine System Storage to Increase the Swap Partition \(Junos OS\) | 28](#)

RELATED DOCUMENTATION

[Recovery Using an Emergency Boot Device \(Junos OS\) | 334](#)

<https://support.juniper.net/support/downloads/>

Install Software on Switches

IN THIS CHAPTER

- Installing Software on EX Series Switches | 143
- Configuring Dual-Root Partitions | 161
- Troubleshooting Software Installation on EX Series Switches | 165
- Troubleshooting a Switch That Has Booted from the Backup Junos OS Image | 170
- Boot EX4000 using Network Boot | 172
- Installing Software on QFX Series Devices (Junos OS) | 174
- Installing and Recovering Software Using the Open Network Install Environment (ONIE) | 191

Installing Software on EX Series Switches

IN THIS SECTION

- Understanding Software Installation on EX Series Switches | 144
- Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine (CLI Procedure) | 146
- Installing Software on an EX Series Switch with Redundant Routing Engines (CLI Procedure) | 149
- Upgrading the Loader Software on the Line Cards in a Standalone EX8200 Switch or an EX8200 Virtual Chassis | 156
- Booting an EX Series Switch Using a Software Package Stored on a USB Flash Drive | 160

EX Series devices are delivered with pre-installed Junos operating system (Junos OS). Before you start this procedure, decide which software package you need and download it. For information on which packages to use for which upgrades, see ["Junos OS and Junos OS Evolved Installation Package Names" on page 15](#).



NOTE: If you are upgrading or installing Junos OS on a VM host, see ["Installing, Upgrading, Backing Up, and Recovery of VM Host"](#) on page 385.

Understanding Software Installation on EX Series Switches

IN THIS SECTION

- [Overview of the Software Installation Process | 144](#)
- [Installing Software on a Virtual Chassis | 145](#)
- [Installing Software Using Automatic Software Download | 145](#)
- [Autoinstalling a Configuration File on an EX2200 or EX3300 Switch from a Disk-on-Key USB Memory Stick | 145](#)
- [Installing Software on an EX2300, EX3400, or EX4100 Switch | 145](#)

A Juniper Networks EX Series Ethernet Switch is delivered with the Juniper Networks Junos operating system (Junos OS) pre-installed. As new features and software fixes become available, you must upgrade your software to use them. You can also downgrade Junos OS to a previous release.

This topic covers:

Overview of the Software Installation Process

An EX Series switch is delivered with a domestic version of Junos OS pre-installed. When you connect power to the switch, it starts (boots) from the installed software.

You upgrade Junos OS on an EX Series switch by copying a software package to your switch or another system on your local network, then use either the J-Web interface or the command-line interface (CLI) to install the new software package on the switch. Finally, you reboot the switch; it boots from the upgraded software. After a successful upgrade, you should back up the new current configuration to a secondary device. You should follow this procedure regardless of whether you are installing a domestic or controlled Junos OS package.

During a successful upgrade, the upgrade package removes all files from **/var/tmp** and completely reinstalls the existing software. It retains configuration files, and similar information, such as secure shell and host keys, from the previous version. The previous software package is preserved in a separate disk partition, and you can manually revert back to it if necessary. If the software installation fails for any reason, such as loss of power during the installation process, the system returns to the originally active installation when you reboot.

Installing Software on a Virtual Chassis

You can connect individual EX Series switches together to form one unit and manage the unit as a single device, called a *Virtual Chassis*. The Virtual Chassis operates as a single network entity composed of member switches. Each member switch in a Virtual Chassis must be running the same version of Junos OS.

For ease of management, a Virtual Chassis provides flexible methods to upgrade software releases. You can deploy a new software release to all member switches of a Virtual Chassis or to only a particular member switch.

You can also upgrade the software on a Virtual Chassis using nonstop software upgrade (NSSU). NSSU takes advantage of *graceful Routing Engine switchover* (GRES) and *nonstop active routing* (NSR) to ensure no disruption to the control plane during the upgrade. You can minimize disruption to network traffic by defining link aggregation groups (LAGs) such that the member links of each LAG reside on different line cards or on different members. During an NSSU, the line cards and Virtual Chassis members are upgraded one at a time, so that traffic continues to flow through the other line cards or members while that line card or member is being upgraded.

Installing Software Using Automatic Software Download

The automatic software download feature uses the DHCP message exchange process to download and install software packages. Users can define a path to a software package on the DHCP server, and then the DHCP server communicates this path to EX Series switches acting as DHCP clients as part of the DHCP message exchange process. The DHCP clients that have been configured for automatic software download receive these messages and, when the software package name in the DHCP server message is different from that of the software package that booted the DHCP client switch, download and install the software package. See ["Upgrading Software by Using Automatic Software Download for Switches" on page 182](#).

Autoinstalling a Configuration File on an EX2200 or EX3300 Switch from a Disk-on-Key USB Memory Stick

You can use an autoinstallation process to configure the software on an EX2200 or EX3300 switch. You can use a configuration file that is in either text format or XML format. If you want to use an XML-formatted file, you use a Junos Space platform to create the configuration file. You place the configuration file on a Disk-on-Key USB memory stick.

Installing Software on an EX2300, EX3400, or EX4100 Switch

Before installing software on an EX2300, EX3400, or EX4100 switch:

- Ensure that at least 620 MB of disk space is available in the system before downloading the software installation package to the `/var/tmp` directory. Use the command `show system storage` to get details of the available space.
- If the space available is inadequate, use the command `request system storage cleanup`. Additionally, you can manually delete any other log or unwanted files from the `/var/tmp` or `/var/log` directories.

You can now follow the procedure in "[Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine \(CLI Procedure\)](#)" on page 146 to complete the software installation.



NOTE: See the [Knowledge Base](#) for more information in regards to storage when upgrading Junos OS on EX2300 and EX3400 switches.

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

You can use this procedure to upgrade Junos OS on a single routing engine in any EX Series switch, including all switches that do not support redundant Routing Engines. You can also use this procedure to upgrade software on all EX Series Virtual Chassis, with the exception of the EX8200 Virtual Chassis.

This procedure can be used to upgrade the following switches or Virtual Chassis:

- EX2200 switch
- EX2300 switch
- EX3200 switch
- EX3300 switch
- EX3400 switch
- EX4100 switch
- EX4200 switch
- EX4300 switch
- EX4500 switch
- EX4550 switch
- EX6200 switch (single Routing Engine upgrade only)
- EX8200 switch (single Routing Engine upgrade only)
- All Virtual Chassis except EX8200 Virtual Chassis

To upgrade software on an EX6200 or EX8200 switch running two Routing Engines, see ["Installing Software on an EX Series Switch with Redundant Routing Engines \(CLI Procedure\)" on page 149](#) or [Understanding Nonstop Software Upgrade on EX Series Switches](#).

To upgrade software on an EX8200 Virtual Chassis, see [Installing Software for All Devices in an EX8200 Virtual Chassis](#).

To install software upgrades on a switch with a single Routing Engine:

1. Download the software package.
2. (Optional) Back up the current software configuration to a second storage option. See the [Backing Up an Installation Using Snapshots \(Junos OS\)](#) for instructions on performing this task.
3. (Optional) Copy the software package to the switch. We recommend that you use FTP to copy the file to the `/var/tmp` directory.

This step is optional because Junos OS can also be upgraded when the software image is stored at a remote location. These instructions describe the software upgrade process for both scenarios.

4. Install the new package on the switch:

```
user@switch> request system software add package
```



NOTE: On EX4300-MP devices, you must use the **force-host** option. For example, issue the **request system software add *source* force-host** command.

Replace *package* with one of the following paths:

- For a software package in a local directory on the switch—`/var/tmp/package.tgz`.
- For a software package on a remote server:
 - `ftp://hostname/pathname/package.tgz`
 - `http://hostname/pathname/package.tgz`

where *package.tgz* is, for example, `jinstall-ex-4200-9.4R1.8-domestic-signed.tgz`.



NOTE: Include the optional **member** option to install the software package on only one member of a Virtual Chassis:

```
user@switch> request system software add source member member-id
```



NOTE: On EX4300-MP devices, you must use the **force-host** option. For example, issue the **request system software add source member member-id force-host** command.

Other members of the Virtual Chassis are not affected. To install the software on all members of the Virtual Chassis, do not include the `member` option.



NOTE: To terminate the installation, do not reboot your device; instead, finish the installation and then issue the `request system software delete package.tgz` command, where `package.tgz` is, for example, **jinstall-ex-4200-10.2R1.8-domestic-signed.tgz**. This is your last chance to stop the installation.

The `request system software delete package.tgz` command is not available on EX2300 and EX3400 switches.

5. Reboot to start the new software:

```
user@switch> request system reboot
```

6. After the reboot has completed, log in and verify that the new version of the software is properly installed:

```
user@switch> show version
```

7. To ensure that the resilient dual-root partitions feature operates correctly, execute the following command to copy the new Junos OS image into the alternate root partition:

```
user@switch> request system snapshot slice alternate
```

To update the alternate root partitions on all members of a Virtual Chassis, use this command:

```
user@switch> request system snapshot slice alternate all-members
```

Resilient dual-root partitions allow the switch to boot transparently from the alternate root partition if the system fails to boot from the primary root partition.



NOTE: EX2300 and EX3400 switches have two volumes: **JUNOS** volume and **OAM (recovery)** volume. To store a snapshot (non-recovery) on JUNOS volume, use the

command request system snapshot. To create snapshot (recovery) on the OAM volume, use the command request system snapshot recovery.

Installing Software on an EX Series Switch with Redundant Routing Engines (CLI Procedure)

IN THIS SECTION

- [Preparing the Switch for the Software Installation | 150](#)
- [Installing Software on the Backup Routing Engine | 152](#)
- [Installing Software on the Default Primary Routing Engine | 153](#)
- [Returning Routing Control to the Default Primary Routing Engine \(Optional\) | 155](#)

You can install software on a switch with redundant Routing Engines in one of two ways:

- Perform an NSSU—An NSSU upgrades both Routing Engines with a single command and with a minimum of network disruption. An NSSU takes advantage of GRES and NSR to ensure no disruption to the control plane. You can minimize disruption to network traffic by defining LAGs such that the member links of each LAG reside on different line cards. The line cards are upgraded one at a time, so that traffic continues to flow through the other line cards while a line card is being upgraded.

You cannot use NSSU to downgrade the software running on a switch.

For more information about NSSU, see *Understanding Nonstop Software Upgrade on EX Series Switches*.

- Upgrade each Routing Engine manually—You can perform a Junos OS installation on each Routing Engine separately, starting with the backup Routing Engine. You can use this procedure to downgrade the software running on a switch.

For an EX6200 switch or an EX8200 switch with redundant Routing Engines, you can minimize disruption to network operation during a Junos OS upgrade by upgrading the Routing Engines separately, starting with the backup Routing Engine.



NOTE: If your EX8200 switch is running Junos OS Release 10.4R3 or later, you can upgrade the software packages on both Routing Engines with a single command and with minimal network disruption by using nonstop software upgrade (NSSU) instead of this procedure. See *Understanding Nonstop Software Upgrade on EX Series Switches*.



CAUTION: If graceful routing engine switchover (GRES) or nonstop active routing (NSR) is enabled when you initiate a software installation, the software does not install properly. Make sure you disable GRES before you begin the software installation by using the `deactivate chassis redundancy graceful-switchover` command in configuration mode. If GRES is enabled, it will be removed with the `redundancy` command. By default, NSR is disabled. If NSR is enabled, remove the `nonstop-routing` statement from the `[edit routing-options]` hierarchy level to disable it.

To upgrade the software package on an EX6200 switch or an EX8200 switch with one installed Routing Engine, see ["Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine \(CLI Procedure\)" on page 146](#).

To upgrade redundant Routing Engines, you first install the new Junos OS release on the backup Routing Engine while keeping the currently running software version on the primary Routing Engine. After making sure that the new software version is running correctly on the backup Routing Engine, you switch device control to the backup Routing Engine. Finally, you install the new software on the new backup Routing Engine.

To upgrade Junos OS on the switch, perform the following tasks:

Preparing the Switch for the Software Installation

Perform the following steps before installing the software:

1. Log in to the primary Routing Engine's console.

For information on logging in to the Routing Engine through the console port, see *Connecting and Configuring an EX Series Switch (CLI Procedure)*.

2. Enter the Junos OS CLI configuration mode:

- a. Start the CLI from the shell prompt:

```
user@switch:RE% cli
```

You will see:

```
{master}
user@switch>
```

- b.

Enter configuration mode:

```
user@switch> configure
```

You will see:

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

3. Disable nonstop active routing (NSR):

```
{master}[edit]  
user@switch# delete routing-options nonstop-routing
```

4. Disable nonstop bridging:

```
{master}[edit]  
user@switch# delete protocols layer2-control nonstop-bridging
```

5. Disable graceful Routing Engine switchover (GRES):

```
{master}[edit]  
user@switch# deactivate chassis redundancy graceful-switchover
```

6. Save the configuration change on both Routing Engines:

```
{master}[edit]  
user@switch# commit synchronize
```



NOTE: To ensure the most recent configuration changes are committed before the software upgrade, perform this step even if nonstop active routing and graceful Routing Engine switchover were previously disabled.

7. Exit the CLI configuration mode:

```
[edit]
user@switch# exit
```

8. (Optional) Back up the current software configuration to a second storage option. See the [Backing Up an Installation Using Snapshots \(Junos OS\)](#) for instructions on performing this task.

Installing Software on the Backup Routing Engine

After you have prepared the switch for software installation, install the software on the backup Routing Engine. During the installation, the primary Routing Engine continues operations, minimizing the disruption to network traffic.

1. Download the software.
2. Copy the software package to the switch. We recommend that you use FTP to copy the file to the `/var/tmp` directory.
3. Log in to the console of the backup Routing Engine.
4. Install the new software package:

```
user@switch> request system software add /var/tmp/package.tgz
```

where *package.tgz* is, for example, **junos-install-ex-arm-64-22.2R1.3.tgz**.



NOTE: To terminate the installation, do not reboot your device; instead, finish the installation and then issue the **request system software delete *package.tgz*** command, where *package.tgz* is, for example, **junos-install-ex-arm-64-22.2R1.3.tgz**. This is your last chance to stop the installation.

5. Reboot to start the new software:

```
user@switch> request system reboot
Reboot the system? [yes, no] (no) yes
```



NOTE: You must reboot the switch to load the new installation of the Junos OS.

6. After the reboot has completed, log in and verify the new version of the software is properly installed:

```
user@switch> show version
```

Installing Software on the Default Primary Routing Engine

To transfer control to the backup Routing Engine and then upgrade or downgrade the primary Routing Engine software:

1. Log in to the primary Routing Engine console port.
2. Transfer control to the backup Routing Engine:



CAUTION: Because graceful Routing Engine switchover is disabled, this switchover causes all line cards in the switch to reload. All network traffic passing through these line cards is lost during the line card reloads.

```
user@switch> request chassis routing-engine master switch
```

3. Verify that the default backup Routing Engine (shown as slot 1 in the command output) is now the primary Routing Engine:

```
user@switch> show chassis routing-engine
```

You will see:

```
Routing Engine status:
```

```
Slot 0:
```

Current state	Backup
Election priority	Master (default)

```
Routing Engine status:
```

```
Slot 1:
```

Current state	Master
Election priority	Backup (default)

4. Install the new software package:

```
user@switch> request system software add package.tgz
```

5. Reboot the Routing Engine:

```
user@switch> request system reboot  
Reboot the system? [yes, no] (no) yes
```

When the reboot completes, the prompt will reappear. Wait for this prompt to reappear before proceeding to the next step.

6. Log in to the default backup Routing Engine (slot 1) through the console port.
7. Re-enable graceful Routing Engine switchover:

```
[edit]  
user@switch# activate chassis redundancy graceful-switchover
```

Re-enabling graceful Routing Engine switchover allows any future Routing Engine switchovers to occur without loss of any network traffic.

8. Re-enable nonstop active routing:

```
[edit]  
user@switch# set routing-options nonstop-routing
```



NOTE: Automatic commit synchronization is a requirement for nonstop active routing. If you have not yet enabled it, do so with the `set system commit synchronize` command.

9. Save the configuration change:

```
[edit]  
user@switch# commit synchronize
```

10. To ensure that the resilient dual-root partitions feature operates correctly, execute the following command to copy the new Junos OS image into the alternate root partition on each Routing Engine:

```
user@switch> request system snapshot slice alternate routing-engine both
```

Resilient dual-root partitions allow the switch to boot transparently from the alternate root partition if the system fails to boot from the primary root partition.

If you want to return routing control to the Routing Engine that was the primary Routing Engine at the beginning of the procedure (the default primary Routing Engine), perform the next task.

Returning Routing Control to the Default Primary Routing Engine (Optional)

The switch can maintain normal operations with the Routing Engine in slot 1 acting as the primary Routing Engine after the software upgrade, so only perform this task if you want to return routing control to the default primary Routing Engine in slot 0.

1. Transfer routing control back to the default primary Routing Engine:

```
user@switch> request chassis routing-engine master switch
```

2. Verify that the default primary Routing Engine (slot 0) is indeed the primary Routing Engine:

```
user@switch> show chassis routing-engine
```

You will see:

```
Routing Engine status:
  Slot 0:
    Current state           Master
    Election priority       Master (default)

Routing Engine status:
  Slot 1:
    Current state           Backup
    Election priority       Backup (default)
```

Upgrading the Loader Software on the Line Cards in a Standalone EX8200 Switch or an EX8200 Virtual Chassis

You are almost never required to upgrade the loader software on the line cards in an EX8200 switch.

Upgrading the loader software version for a line card is not a requirement to complete any software upgrade. In rare cases, a line card might go offline immediately after a software upgrade because the loader software version on the line card requires an upgrade to become compatible with the upgraded Junos OS. You can upgrade the loader software on the line cards as a best practice to avoid this problem and other less severe issues.

The loader software on any line card in an EX8200 switch is updated using the same loader software package that upgrades the EX8200 Routing Engine loader software. The line card software loader contains two banks, each with a single loader software version. This procedure is used to upgrade the loader software for both banks of a line card in a standalone EX8200 switch or an EX8200 Virtual Chassis.

To upgrade the loader software on the line cards in a standalone EX8200 switch or an EX8200 Virtual Chassis:



NOTE: If you are upgrading Junos OS, the Routing Engine loader software, and the line card loader software, we recommend that you upgrade in this order: Junos OS, line card loader software, Routing Engine loader software.

1. Determine the version of the loader software for the line cards:

```
user@switch> show chassis firmware
```

Part	Type	Version
FPC 6	U-Boot	U-Boot 1.1.6 (Jan 13 2009 - 06:55:22) 2.3.0
	loader	FreeBSD/PowerPC U-Boot bootstrap loader 2.2
FPC 7	U-Boot	U-Boot 1.1.6 (Jan 13 2009 - 06:55:22) 2.3.0
	loader	FreeBSD/PowerPC U-Boot bootstrap loader 2.2
Routing Engine 0	U-Boot	U-Boot 1.1.6 (Mar 11 2011 - 04:29:01) 3.5.0
	loader	FreeBSD/PowerPC U-Boot bootstrap loader 2.4
Routing Engine 1	U-Boot	U-Boot 1.1.6 (Mar 11 2011 - 04:29:01) 2.3.0
	loader	FreeBSD/PowerPC U-Boot bootstrap loader 2.4



NOTE: On an EX8200 Virtual Chassis, you cannot issue the `show chassis firmware` command on the primary external Routing Engine. You must issue this command on each member switch.

- a. From the primary external Routing Engine, start a shell session on the member switch, for example:

```
user@external-routing-engine> request session member 0
```

- b. Enter the CLI and issue the `show chassis firmware` command.
- c. Repeat these steps for the other member switch.

The loader software version appears after the timestamp (see the `Version` column in the output) for each component. For example, In the example given in this step, look at the first FPC listed (FPC 6). Ignore the U-Boot version number (1.1.6) and find the loader software version number (2.3.0) after the timestamp (U-Boot 1.1.6 (Jan 13 2009 - 06:55:22)). The U-Boot version number has nothing to do with the loader software version that you need to determine.

If the loader software version is earlier than 3.5.0 for any FPC, you should consider upgrading the loader software for that line card.

2. Download the loader software package from the Juniper Networks Download page (<https://support-www.juniper.net/support/downloads/>) and place the software package on an internal software distribution site or in a local directory on the switch. We recommend using `/var/tmp` as the local directory on the switch.



NOTE: To obtain the loader software package, see the Download Software page at <https://support-www.juniper.net/support/downloads/>. Select the OS type and the release. Then find and click the download image.

A login screen appears.

3. Log in with your user name and password.
4. Disable graceful Routing Engine switchover (GRES) and nonstop active routing (NSR), if enabled. Commit the configuration:

```
user@switch# deactivate chassis redundancy graceful-switchover
user@switch# deactivate routing-options nonstop-routing
user@switch# commit synchronize
```

5. Install the loader package:

```
user@switch> request system software add package
```

Replace *package* with one of the following paths:

- For a software package in the `/var/tmp` directory on the switch or external Routing Engine —`/var/tmp/package.tgz`.
- For a software package on a remote server:
 - `ftp://hostname/pathname/package.tgz`
 - `http://hostname/pathname/package.tgz`

In the above options, *package.tgz* might be, for example, `jloader-ex-8200-11.3build-signed.tgz`.

6. Upgrade the loader software.

- To upgrade the loader software for a line card on a standalone EX8200 switch:

```
user@switch> request system firmware upgrade fpc slot slot-number
Firmware upgrade initiated....
Please wait for ~2mins for upgrade to complete....
```

- To upgrade the loader software for a line card on an EX8200 member switch in an EX8200 Virtual Chassis:

```
user@switch> request system firmware upgrade fpc slot slot-number member member-id
Firmware upgrade initiated....
Please wait for ~2mins for upgrade to complete....
```

7. Confirm the loader software upgrade:

```
user@switch> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
FPC 6	U-Boot	0	2.3.0		UPGRADED SUCCESSFULLY
FPC 7	U-Boot	0	2.3.0		OK
Routing Engine 0	RE BIOS	0	3.1.1		OK
Routing Engine 1		0	3.1.1		OK

The status is `UPGRADED SUCCESSFULLY` if the boot loader version update process is complete.

The status is `PROGRAMMING` if the boot loader version update process is still in progress.

Do not proceed to the next step until the `show system firmware` output confirms that the loader software upgrade is complete.

8. Restart the line card.

- To restart a line card on a standalone EX8200 switch:

```
user@switch> request chassis fpc restart slot slot-number
```

- To restart a line card on an EX8200 member switch in an EX8200 Virtual Chassis:

```
user@switch> request chassis fpc restart slot slot-number member member-id
```



NOTE: You can monitor the status of the line card restart by using the `show chassis fpc` command.

9. After the line card restart has completed, confirm the loader software version update:

```
user@switch> show chassis firmware
```

Part	Type	Tag	Current version	Available version	Status
FPC 6	U-Boot	0	3.5.0		OK
FPC 7	U-Boot	0	2.3.0		OK
Routing Engine 0 RE BIOS		0	3.1.1		OK
Routing Engine 1		0	3.1.1		OK

The current version has updated to 3.5.0. You have upgraded the loader software for one bank of the line card.

10. Repeat Steps 4 through 7 to upgrade the loader software on the other bank of the line card.



NOTE: A bank switchover occurs automatically as part of the line card restart. Repeating Steps 3 through 6 updates the loader software on the other bank.

11. Repeat Steps 4 through 8 for all other line cards that require a line card loader version upgrade.

SEE ALSO

[Understanding Nonstop Software Upgrade on EX Series Switches](#)

Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis (CLI Procedure)

[Understanding Power Management on EX Series Switches](#)

Booting an EX Series Switch Using a Software Package Stored on a USB Flash Drive

There are two methods of getting Junos OS stored on a USB flash drive before using the software to boot the switch. You can pre-install the software onto the USB flash drive before inserting the USB flash drive into the USB port, or you can use the system snapshot feature to copy files from internal switch memory to the USB flash drive.

To move files into USB flash memory by using a system snapshot and use those files to boot the switch, see ["Creating a Snapshot and Using It to Boot an EX Series Switch" on page 317](#). We recommend that you use this method to boot the switch from a USB flash drive if your switch is running properly.

If you need to pre-install the software onto the USB flash drive, you can use the method described in this topic. Pre-installing Junos OS onto a USB flash drive to boot the switch can be done at any time and is particularly useful when the switch boots to the loader prompt because the switch cannot locate the Junos OS in internal flash memory.

Ensure that you have the following tools and parts available to boot the switch from a USB flash drive:

- A USB flash drive that meets the EX Series switch USB port specifications. See [USB Port Specifications for an EX Series Switch](#).
- A computer or other device that you can use to download the software package from the Internet and copy it to the USB flash drive.

To download a Junos OS package onto a USB flash drive before inserting the USB flash drive:

1. Download the Junos OS package that you want to place onto the EX Series switch from the Internet onto the USB flash drive by using your computer or other device.
2. Remove the USB flash drive from the computer or other device.
3. Insert the USB flash drive into the USB port on the switch.
4. This step can be performed only when the prompt for the loader script (loader>) is displayed. The loader script starts when the Junos OS loads but the CLI is not working for any reason or if the switch has no software installed.

Install the software package onto the switch:

```
loader> install source
```


where *source* represents the name and location of the Junos OS package on the USB flash drive. The Junos OS package on a flash drive is commonly stored in the root drive as the only file—for example, `file:///junos-install-media-usb-ex-arm-64-22.2R1.3.tgz`.

RELATED DOCUMENTATION

[Understanding Nonstop Software Upgrade on EX Series Switches](#)

[Troubleshooting Software Installation on EX Series Switches](#) | 165

Configuring Dual-Root Partitions

IN THIS SECTION

- [Resilient Dual-Root Partition Scheme \(Junos OS Release 10.4R3 and Later\)](#) | 162
- [Automatic Fixing of Corrupted Primary Root Partition with the Automatic Snapshot Feature](#) | 162
- [Earlier Partition Scheme \(Junos OS Release 10.4R2 and Earlier\)](#) | 164
- [Understanding Upgrading or Downgrading Between Resilient Dual-Root Partition Releases and Earlier Releases](#) | 164

Resilient dual-root partitioning, introduced on Juniper Networks EX Series Ethernet Switches in Juniper Networks Junos operating system (Junos OS) Release 10.4R3, provides additional resiliency to switches in the following ways:

- Allows the switch to boot transparently from the second (alternate) root partition if the system fails to boot from the primary root partition.
- Provides separation of the root Junos OS file system from the `/var` file system. If corruption occurs in the `/var` file system (a higher probability than in the root file system because of the greater frequency of reads and writes in `/var`), the root file system is insulated from the corruption.



NOTE: For instructions on upgrading to a release that supports resilient dual-root partitions from a release that does not, see the release notes. The procedure for upgrading to a resilient dual-root partition release is different from the normal upgrade procedure.

Resilient Dual-Root Partition Scheme (Junos OS Release 10.4R3 and Later)

EX Series switches that ship with Junos OS Release 10.4R3 or later are configured with a root partition scheme that is optimized for resiliency, as shown in [Table 14 on page 162](#).

Table 14: Resilient Dual-Root Partition Scheme

Slice 1	Slice 2	Slice 3		Slice 4
s1a	s2a	s3e	s3d	s4d
/	/	/var	/var/tmp	/config
(root Junos OS)	(root Junos OS)			

In the resilient dual-root partition scheme, the **/var** file system is contained in a separate slice (Slice 3) from the root file systems, the **/config** directory is contained in its own slice (Slice 4), and switches ship from the factory with identical Junos OS images in Slice 1 and Slice 2. The **/var** file system, which has a greater frequency of reads and writes than the root file systems and is therefore more likely to have corruption issues, is isolated from the root directories and the **/config** directory. If the switch fails to boot from the active partition, the switch automatically boots from the alternate root partition and triggers an alarm.

Automatic Fixing of Corrupted Primary Root Partition with the Automatic Snapshot Feature

Resilient dual-root partitioning also provides the *automatic snapshot* feature, which allows the switch to automatically fix a corrupt Junos OS file in the primary root partition. If the automatic snapshot feature is enabled, the switch automatically takes a snapshot of the Junos OS root file system in the alternate root partition and copies it onto the primary root partition, thereby repairing the corrupt file in the primary root partition. The automatic snapshot procedure takes place whenever the system reboots from the alternate root partition, regardless of whether the reboot is due to a command or due to corruption of the primary root partition.



NOTE:

- EX9200 switches do not support the automatic snapshot feature.

- The automatic snapshot feature is enabled by default on the following EX Series switches:
 - EX4550 switches
 - EX Series switches that ship with Junos OS Release 12.3R1 or later
- The automatic snapshot feature is disabled by default on EX Series switches (except the EX4550 switches) running Junos OS Release 12.2 or earlier.
- If the automatic snapshot feature was disabled by default before the switch was upgraded to Junos OS Release 12.3R1 or later, the feature remains disabled (for backward compatibility) by default after the upgrade.
- If the automatic snapshot feature is enabled in a *Virtual Chassis* configuration, the automatic snapshot procedure takes place whenever any member of the Virtual Chassis reboots from its alternate root partition.
- You can enable the automatic snapshot feature by configuring the `auto-snapshot` statement at the `[edit system]` hierarchy level.

The automatic snapshot feature provides an additional layer of fault protection if you maintain the same version of Junos OS in both partitions of resilient dual-root partitions. When `auto-snapshot` is enabled, repair happens automatically. Therefore, the switch does not issue an alarm to indicate that the system has rebooted from the alternate partition. However, it does log the event. You cannot execute a manual snapshot when an automatic snapshot procedure is in process. The login banner indicates that an automatic snapshot operation is in progress and that banner is removed only after the snapshot operation is complete. The next reboot happens from the primary partition.



NOTE: EX Series switches that ship with Junos OS Release 10.4R3 or later are configured with identical Junos OS images in the primary root partition (Slice 1) and the alternate root partition (Slice 2).

However, if you do *not* maintain the same version of Junos OS in both partitions, you might want to disable the automatic snapshot feature. If you have an earlier version of Junos OS in the alternate partition and the system reboots from the alternate root partition, the automatic snapshot feature causes the later Junos OS version to be replaced with the earlier version.

When automatic snapshot is disabled and the system reboots from the alternate root partition, it triggers an alarm indicating that the system has rebooted from its alternate partition.

Earlier Partition Scheme (Junos OS Release 10.4R2 and Earlier)

The partition scheme used in Junos OS 10.4R2 and earlier is shown in [Table 15 on page 164](#).

Table 15: Earlier Partition Scheme

Slice 1		Slice 2		Slice 3	
s1a	s1f	s2a	s2f	s3d	s3e
/	/var	(empty until initial software upgrade)	(empty until initial software upgrade)	/var/tmp	/config
(root Junos OS)					

This is the partitioning scheme for a switch shipped with Release 10.4R2 or earlier (or after you reformat the disk during a downgrade from Release 10.4R3 or later to Release 10.4R2 or earlier). In this partitioning scheme, the switch comes from the factory with only one Junos OS image installed in the root Junos OS partition of Slice 1. The first time that you perform a software upgrade, the new Junos OS image is installed in Slice 2. If the switch fails to boot, you must manually trigger it to boot from the alternate partition (rebooting from the alternate partition does not occur automatically).

Understanding Upgrading or Downgrading Between Resilient Dual-Root Partition Releases and Earlier Releases

Upgrading from Release 10.4R2 or earlier to Release 10.4R3 or later differs from other upgrades in two important ways:

- You must install a new loader software package in addition to installing the new Junos OS image.
- Rebooting after the upgrade reformats the disk from three partitions to four partitions. See [Table 14 on page 162](#).

You can perform all operations for this special software upgrade from the CLI.



CAUTION: Back up any important log files because the **/var/log** files are not saved or restored during an upgrade from Release 10.4R2 or earlier to a release that supports resilient dual-root partitions (Release 10.4R3 or later).

We recommend that you also save your **/config** files and any important log files to an external medium because if there is a power interruption during the upgrade process, they might be lost.

RELATED DOCUMENTATION

| *auto-snapshot*

Troubleshooting Software Installation on EX Series Switches

IN THIS SECTION

- [Recovering from a Failed Software Upgrade on an EX Series Switch | 165](#)
- [Rebooting from the Inactive Partition | 167](#)
- [Freeing Disk Space for Software Installation | 168](#)
- [Installation from the Boot Loader Generates 'cannot open package' Error | 169](#)

This topic describes troubleshooting issues with software installations on EX Series switches.

Recovering from a Failed Software Upgrade on an EX Series Switch

IN THIS SECTION

- [Problem | 165](#)
- [Solution | 166](#)

Problem

Description

If Junos OS loads but the CLI is not working, or if the switch has no software installed, use this recovery installation procedure to install Junos OS.

Solution

If there is already a Junos OS image on the system, you can either install the new Junos OS package in a separate partition and have both Junos OS images remain on the system, or you can wipe the disk clean before the new installation proceeds.

If there is no Junos OS image on the system, follow the instructions in [Booting an EX Series Switch Using a Software Package Stored on a USB Flash Drive](#) to get an image on the system and boot the switch.

To perform a recovery installation:

1. Power on the switch.

The loader script starts.

After the message `Loading /boot/defaults/loader.conf` displays, you are prompted with the following:

```
Hit [Enter] to boot immediately, or space bar for command prompt.
```

2. Press the space bar to enter the manual loader.

The `loader>` prompt displays.

3. Enter the following command:

```
loader> install [- -format] [- -external] source
```

where:

- **format**—Use this option to wipe the installation media before installing the software package. If you do not include this option, the system installs the new Junos OS package in a different partition from the partition used by the most recently installed Junos OS package.
- **external**—Use this option to install the software package on an external medium.
- **source**—Represents the name and location of the Junos OS package either on a server on the network or as a file on the USB flash drive:
 - Network address of the server and the path on the server; for example, **tftp://192.168.1.28/junos/jinstall-ex-4200-9.4R1.5-domestic-signed.tgz**
 - The Junos OS package on a USB device is commonly stored in the root drive as the only file; for example, **file:///jinstall-ex-4200-9.4R1.5-domestic-signed.tgz**

The boot process proceeds as normal and ends with a login prompt.

Rebooting from the Inactive Partition

IN THIS SECTION

● Problem | 167

● Solution | 167

Problem

Description

EX Series switches shipped with Junos OS Release 10.4R2 or earlier have Junos OS loaded on the system disk in partition 1. The first time you upgrade, the new software package is installed in partition 2. When you finish the installation and reboot, partition 2 becomes the active partition. Similarly, subsequent software packages are installed in the inactive partition, which becomes the active partition when you reboot at the end of the installation process.

On switches shipped with Release 10.4R3 and later, the same Junos OS image is loaded in each of the two root partitions, and you should copy the new software image to the alternate partition each time you upgrade.

If you performed an upgrade and rebooted, the system resets the active partition. You can use this procedure to manually boot from the inactive partition.



NOTE: If you have completed the installation of the software image but have not yet rebooted, issue the `request system software rollback` command to return to the original software installation package.

Solution

Reboot from the inactive partition:

```
user@switch> request system reboot slice alternate
```



NOTE: If you cannot access the CLI, you can reboot from the inactive partition using the following procedure from the loader script prompt:

1. Unload and clear the interrupted boot from the active partition:

```
loader> unload  
loader> unset vfs.root.mountfrom
```

2. Select the new (inactive) partition to boot from:

```
loader> set currdev=diskxsy:
```

where x is either 0 (internal) or 1 (external), and the y indicates the number of the inactive partition, either 1 or 2.

You must include the colon (:) at the end of this command.

3. Boot Junos OS from the inactive partition:

```
loader> boot
```

Freeing Disk Space for Software Installation

IN THIS SECTION

- [Problem | 168](#)
- [Solution | 169](#)

Problem

Description

The software installation process requires a certain amount of unused disk space. If there is not enough space, you might receive an error message such as:

```
fetch: /var/tmp/incoming-package.tgz: No space left on device
```


Solution

Identify and delete unnecessary files by using the `request system storage cleanup` command.

Installation from the Boot Loader Generates 'cannot open package' Error

IN THIS SECTION

● [Problem | 169](#)

● [Solution | 169](#)

Problem

Description

When installing a Junos OS software image from the loader prompt, a “cannot open package error” is generated:

```
loader> install - -format
tftp://10.204.33.248/images/Flash_corr/official/jinstall-ex-4200-10.4I2011012-domestic-signed.tgz
Speed: 1000, full duplex
bootp: no reply
No response for RARP request
net_open: RARP failed
cannot open package (error 5)
```

Solution

This might be due to the IP address, gateway IP address, netmask address, or server IP address not being properly set. You can set these values either from the shell or from the u-boot prompt.

To set these values from the shell:

```
% nvrw setenv ipaddr 10.204.35.235
% nvrw setenv netmask 255.255.240.0
% nvrw setenv gatewayip 10.204.47.254
% nvrw setenv serverip 10.204.33.248
```

To set these values from the u-boot prompt, log in to a console connection, reboot, and stop at the u-boot prompt (Cntrl+c):

```
=> setenv ipaddr 10.204.35.235
=> setenv gatewayip 10.204.47.254
=> setenv serverip 10.204.33.248
=> setenv netmask 255.255.240.0
=> saveenv
=> printenv Verify whether variables are set properly or not
=> boot
```

RELATED DOCUMENTATION

[Installing Software on an EX Series Switch with a Virtual Chassis or Single Routing Engine \(CLI Procedure\) | 146](#)

[Upgrading Software on an EX6200 or EX8200 Standalone Switch Using Nonstop Software Upgrade \(CLI Procedure\)](#)

[Installing Software on EX Series Switches \(J-Web Procedure\)](#)

[Understanding Software Installation on EX Series Switches | 144](#)

show system storage partitions

Troubleshooting a Switch That Has Booted from the Backup Junos OS Image

IN THIS SECTION

- [Problem | 171](#)
- [Solution | 171](#)

Problem

Description

The switch boots from the backup root file partition. It is possible that the primary copy of Junos OS failed to boot properly, which could indicate that it is corrupted. This event is flagged in two ways:

- Upon login through the console or management port, the following warning message is displayed:

```
WARNING: THIS DEVICE HAS BOOTED FROM THE BACKUP JUNOS IMAGE
```

It is possible that the primary copy of JUNOS failed to boot up properly, and so this device has booted from the backup copy.

Please re-install JUNOS to recover the primary copy in case it has been corrupted.

- The following alarm message is generated:

```
user@switch> show chassis alarms
1 alarms currently active
Alarm time          Class  Description
2011-02-17 05:48:49 PST  Minor  Host 0 Boot from backup root
```

If the switch is in a Virtual Chassis, the switch member number appears in the Description field, where the switch is called a host.

Solution

Install a new Junos OS image on the partition that had the corruption, or take a snapshot (use *request system snapshot*) of the currently active partition and use it to replace the image in the alternate partition:

If the switch is a standalone switch or a Virtual Chassis primary switch, enter this command:

```
user@switch> request system snapshot slice alternate
```

If the switch is a Virtual Chassis member switch (not the primary), enter this command on the Virtual Chassis:

```
user@switch> request system snapshot slice alternate member member-id
```

where *member-id* is the Virtual Chassis member ID number.

RELATED DOCUMENTATION

[Verifying Junos OS and Boot Loader Software Versions on an EX Series Switch | 66](#)

[Troubleshooting Software Installation on EX Series Switches | 165](#)

show system storage partitions

Boot EX4000 using Network Boot

SUMMARY

Network booting (netboot), refers to the process of initiating a device's startup directly from a network source, rather than relying on local storage devices such as hard drives or USB drives. This method enables the device to load the operating system from a centralized server over the network.

IN THIS SECTION

- [Use Network Boot | 173](#)

Network booting (netboot) is the process of starting the device from a network location instead of a local storage device like a hard disk or a USB drive. Netboot allows the device to boot from a central server and load the Junos OS over the network.

Administrators can efficiently update or upgrade the device firmware across the network using a central server, ensuring all devices are equipped with the latest features and security patches. Since the firmware is derived directly from a central server, the risk of tampering or exploitation is vastly reduced. Furthermore, centralized logging and monitoring of firmware and configuration changes provide a reliable audit trail, making it easier to detect any unauthorized modifications.

The device is configured to initially attempt loading boot files from the local hard disk, failing which it attempts to load the files from the USB. In case, the USB is not available or if there is any other form of failure, then the system attempts netboot. You can also manually select the netboot option (BOOT0001 NET00) on the console when the device restarts, irrespective of the status of the hard disk and USB.

Use Network Boot

This section provides the steps to use network boot.

Prerequisites

The DHCP server should be configured to provide:

- the TFTP server information as the next-hop server
- the loader file location in the TFTP server

Follow the steps below to boot EX4000 using network boot.

After a power cycle, the following options are shown in the U-Boot Boot Menu.

```
*** U-Boot Boot Menu ***
```

```
BOOT0000 HDD00
```

```
BOOT0000 USB00
```

```
BOOT0001 NET00
```

```
U-Boot console
```

```
Press UP/DOWN to move, ENTER to select, ESC/CTRL+C to quit
```

1. Select the BOOT0001 NET00 option and press **Enter**.



NOTE: If you do not select the option manually, the device's U-Boot is configured to automatically look for a bootable image on the network after failing to boot from hard disk and USB.

The following processes are executed automatically on the device:

- a. The `bootp` command is executed by the system which sends a request to a DHCP server to obtain the IP address and other network configuration information.
- b. The DHCP server responds with the IP address of the network boot server, which contains the loader filename.
- c. The device downloads the loader file and boots using the file.
- d. After booting, the console displays the OK prompt.

2. Run the following TFTP install command, where `junos-install-media-net-ex40xx.tgz` is the filename of the image you want to install.

```
OK install tftp://192.168.65.31/junos-install-media-net-ex40xx-.tgz
```

The device then connects to the provided server location, downloads the necessary files using TFTP and then installs the image securely.

Installing Software on QFX Series Devices (Junos OS)

IN THIS SECTION

- [Installing Software Packages on QFX Series Devices \(Junos OS\) | 174](#)
- [Upgrading Software by Using Automatic Software Download for Switches \(Junos OS\) | 182](#)
- [Installing Junos OS Software with Junos Automation Enhancements | 185](#)

QFX Series devices are delivered with the Junos operating system (Junos OS) preinstalled. Before you start this procedure, decide which software package you need and download it. For information on which packages to use for which upgrades, see "[Junos OS Installation Package Names](#)" on [page 15](#).

Installing Software Packages on QFX Series Devices (Junos OS)

IN THIS SECTION

- [Installing the Software on QFX10002-60C Switches | 175](#)
- [Installing a Standard Software Package on QFX5000 and EX4600 Switches | 176](#)
- [Installing a Standard Software Package on QFX10002 Switches | 177](#)
- [Installing a Software Package on QFX10008 and QFX10016 Switches | 178](#)

We recommend that you connect to the console port while installing the installation package so you can respond to any required user input and detect any errors that may occur.

Before you install the new installation package, back up your current configuration files because the upgrade process removes all of the stored files on the switch.

To back up your current configuration files, enter the `save` command:

```
user@switch# save filename
```

Executing this command saves a copy of your configuration files to a remote location such as an external USB device.

Installation procedures are in the following subsections:

Installing the Software on QFX10002-60C Switches

This section explains how to upgrade the software, which includes both the host OS and the Junos OS. This upgrade requires that you use a VM host package—for example, a `junos-vmhost-install-x.tgz`.

During a software upgrade, the alternate partition of the SSD is upgraded, which will become primary partition after a reboot. If there is a boot failure on the primary SSD, the switch can boot using the snapshot available on the alternate SSD.



NOTE: The QFX10002-60C switch supports only the 64-bit version of Junos OS.



NOTE: If you have important files in directories other than `/config` and `/var`, copy the files to a secure location before upgrading. The files under `/config` and `/var` (except `/var/etc`) are preserved after the upgrade.

To upgrade the software, you can use the following methods:

1. If the installation package resides locally on the switch, issue the **request vmhost software add** `<pathname><source>` command.

For example:

```
user@switch> request vmhost software add /var/tmp/junos-vmhost-install-qfx-x86-64-18.1R1.9.tgz
```

If the Install Package resides remotely from the switch, issue the **request vmhost software add** `<pathname><source>` command.

For example:

```
user@switch> request vmhost software add ftp://ftpserver/directory/junos-vmhost-install-qfx-
x86-64-18.1R1.9.tgz
```

2. After the reboot has finished, verify that the new version of software has been properly installed by issuing the `show version` command.

```
user@switch> show version
```

Installing a Standard Software Package on QFX5000 and EX4600 Switches



NOTE: Before you install the software, back up any critical files in `/var/home`. For more information regarding how to back up critical files, contact Customer Support at <https://www.juniper.net/support>.



NOTE: On QFX5100 and EX4600 switches, the Host OS is not upgraded automatically, so you must use the `force-host` option if you want the Junos OS and Host OS versions to be the same.

However, pay attention to these notes regarding Junos OS and Host OS versions:

- The Junos OS and Host OS versions do not need to be the same.
- During an ISSU, the Host OS cannot be upgraded.
- Upgrading the Host OS is not required for every software upgrade, as noted above.



NOTE: The QFX5100 and EX4600 standalone SKUs and non-mixed Virtual Chassis support software images with the package filenames in the `yyy-qfx-5-zzz` (non-TVP architecture) format, for all Junos OS releases **up to** Junos OS Release **21.4**. They **do not** support software images with the package filenames in the `yyy-qfx-5e-zzz` (TVP architecture) format.

1. If the installation package resides locally on the switch, issue the `request system software add <pathname><source> reboot` command.

For example:

```
user@switch> request system software add /var/tmp/jinstall-host-qfx-5-17.2R1.n-signed.tgz
reboot
```

If the Install Package resides remotely, issue the **request system software add <pathname><source> reboot** command.

For example:

```
user@switch> request system software add ftp://ftpserver/directory/jinstall-host-
qfx-5-17.2R1.n-signed.tgz reboot
```

2. After the reboot has finished, verify that the new version of software has been properly installed by executing the **show version** command.

```
user@switch> show version
```

Installing a Standard Software Package on QFX10002 Switches



NOTE: Before you install the software, back up any critical files in **/var/home**. For more information regarding how to back up critical files, contact Customer Support at <https://www.juniper.net/support>.

1. Install the software in one of two ways:
 - If the installation package resides locally on the switch, issue the **request system software add <pathname><source> reboot** command.

For example:

```
user@switch> request system software add /var/tmp/jinstall-host-qfx-10-17.2R1.n-secure-
signed.tgz reboot
```

- If the Install Package resides remotely, issue the **request system software add <pathname><source> reboot** command.

For example:

```
user@switch> request system software add ftp://ftpserver/directory/jinstall-host-
qfx-10-17.2R1.n-secure-signed.tgz reboot
```

2. After the reboot has finished, verify that the new version of software has been properly installed by issuing the `show version` command.

```
user@switch> show version
```

Installing a Software Package on QFX10008 and QFX10016 Switches

IN THIS SECTION

- Preparing the Switch for Installation (Junos OS) | 179
- Installing Software on the Backup Routing Engine (Junos OS) | 179
- Installing Software on the Primary Routing Engine (Junos OS) | 180

Because the switch has two Routing Engines, perform a Junos OS installation on each Routing Engine separately to avoid disrupting network operation.



NOTE: Before you install the software, back up any critical files in `/var/home`. For more information regarding how to back up critical files, contact Customer Support at <https://www.juniper.net/support>.



CAUTION: If graceful Routing Engine switchover (GRES), nonstop bridging (NSB), or nonstop active routing (NSR) is enabled when you initiate a software installation, the software does not install properly. Make sure you issue the CLI `delete chassis redundancy` command when prompted. If GRES is enabled, it will be removed with the redundancy command. By default, NSR is disabled. If NSR is enabled, remove the nonstop-routing statement from the `[edit routing-options]` hierarchy level to disable it.

To upgrade the software, perform the following tasks:

Preparing the Switch for Installation (Junos OS)

Perform the following steps before installing the software:

1. Log in to the primary Routing Engine's console.

For more information about logging in to the Routing Engine through the console port, see the specific hardware guide for your switch.

2. From the command line, enter configuration mode:

```
user@switch> configure
```

3. Disable Routing Engine redundancy:

```
user@switch# delete chassis redundancy
```

4. Disable nonstop-bridging:

```
user@switch# delete protocols layer2-control nonstop-bridging
```

5. Save the configuration change on both Routing Engines:

```
user@switch# commit synchronize
```

6. Exit the CLI configuration mode:

```
user@switch# exit
```

Installing Software on the Backup Routing Engine (Junos OS)

After the switch has been prepared, you first install the new Junos OS release on the backup Routing Engine, while keeping the currently running software version on the primary Routing Engine. This enables the primary Routing Engine to continue operations, minimizing disruption to your network.

After making sure that the new software version is running correctly on the backup Routing Engine, you are ready to switch routing control to the backup Routing Engine, and then upgrade or downgrade the software version on the other Routing Engine.

1. Log in to the console port on the other Routing Engine (currently the backup).

For more information about logging in to the Routing Engine through the console port, see the specific hardware guide for your switch.

2. Install the new software package using the `request system software add` command:

```
user@switch> request system software add validate /var/tmp/jinstall-host-qfx-10-17.2R1.n-secure-signed.tgz
```

For more information about the `request system software add` command, see the [CLI Explorer](#).

3. Reboot the switch to start the new software using the `request system reboot` command:

```
user@switch> request system reboot
```



NOTE: You must reboot the switch to load the new installation of Junos OS on the switch.

To terminate the installation, do not reboot your switch. Instead, finish the installation and then issue the `request system software delete <package-name>` command. This is your last chance to stop the installation.

All the software is loaded when you reboot the switch. Installation can take between 5 and 10 minutes. The switch then reboots from the boot device on which the software was just installed. When the reboot is complete, the switch displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation is not sending traffic.

4. Log in and issue the `show version` command to verify the version of the software installed.

```
user@switch> show version
```

Installing Software on the Primary Routing Engine (Junos OS)

Once the software is installed on the backup Routing Engine, you are ready to switch routing control to the backup Routing Engine, and then upgrade or downgrade the primary Routing Engine software:

1. Log in to the primary Routing Engine console port.

For more information about logging in to the Routing Engine through the console port, see the specific hardware guide for your switch.

2. Transfer routing control to the backup Routing Engine:

```
user@switch> request chassis routing-engine master switch
```

For more information about the `request chassis routing-engine master` command, see the [CLI Explorer](#).

3. Verify that the backup Routing Engine (slot 1) is the primary Routing Engine:

```
user@switch> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state           Backup
    Election priority       Master (default)

Routing Engine status:
  Slot 1:
    Current state           Master
    Election priority       Backup (default)
```

4. Install the new software package using the `request system software add` command:

```
user@switch> request system software add validate /var/tmp/jinstall-host-qfx-10-17.2R1.n-secure-signed.tgz
```

For more information about the `request system software add` command, see the [CLI Explorer](#).

5. Reboot the Routing Engine using the `request system reboot` command:

```
user@switch> request system reboot
```



NOTE: You must reboot to load the new installation of Junos OS on the switch.

To terminate the installation, do not reboot your system. Instead, finish the installation and then issue the `request system software delete jinstall <package-name>` command. This is your last chance to stop the installation.

The software is loaded when you reboot the system. Installation can take between 5 and 10 minutes. The switch then reboots from the boot device on which the software was just installed. When the reboot is complete, the switch displays the login prompt.

While the software is being upgraded, the Routing Engine on which you are performing the installation does not send traffic.

6. Log in and issue the `show version` command to verify the version of the software installed.

7. Transfer routing control back to the primary Routing Engine:

```
user@switch> request chassis routing-engine master switch
```

For more information about the `request chassis routing-engine master` command, see the [CLI Explorer](#).

8. Verify that the primary Routing Engine (slot 0) is indeed the primary Routing Engine:

```
user@switch> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state           Master
    Election priority       Master (default)

Routing Engine status:
  Slot 1:
    Current state           Backup
    Election priority       Backup (default)
```

Upgrading Software by Using Automatic Software Download for Switches (Junos OS)

IN THIS SECTION

- [Configuring DHCP Services for the Switch \(Junos OS\) | 183](#)
- [Enabling Automatic Software Download on a Switch \(Junos OS\) | 183](#)
- [Verifying That Automatic Software Download Is Working Correctly \(Junos OS\) | 184](#)

The automatic software download feature uses the Dynamic Host Configuration Protocol (DHCP) message exchange process to download and install software packages. You configure the automatic software download feature on switches that act as DHCP clients. You must enable automatic software download on a switch before the software upgrade can occur.

You configure a path to a software package file on the DHCP server. The server communicates the path to the software package file through DHCP server messages.

If you enable automatic software download, the DHCP client switch compares the software package name in the DHCP server message with the name of the software package that booted the switch. If the software packages are different, the DHCP client switch downloads and installs the software package specified in the DHCP server message.

Complete the following tasks in order:

Configuring DHCP Services for the Switch (Junos OS)

Before you upgrade software by using automatic software download, ensure that you have configured DHCP services for the switch, including configuring a path to a boot server and a boot file.

To configure a path to a boot server and a boot file:

1. Configure the name of the boot server advertised to DHCP clients. The client uses a boot file located on the boot server to complete DHCP setup. This configuration is equivalent to DHCP option 66:

```
[edit system services dhcp]
user@switch# set boot-server (address | hostname)
```

2. Set the boot file advertised to DHCP clients. After the client receives an IP address and the boot file location from the DHCP server, the client uses the boot image stored in the boot file to complete the DHCP setup. This configuration is equivalent to DHCP option 67:

```
[edit system services dhcp]
user@switch# set boot-file filename
```

Enabling Automatic Software Download on a Switch (Junos OS)

To enable automatic software download on a switch that acts as a DHCP client:

```
[edit chassis]
user@switch# set auto-image-upgrade
```

After automatic software download is enabled on your DHCP client switch and after DHCP services are enabled on your network, an automatic software download can occur at any time as part of the DHCP message exchange process.

If an automatic software download occurs, you see the following message on the switch:

```
Auto-image upgrade started
On successful installation system will reboot automatically
```

The switch reboots automatically to complete the upgrade.

Verifying That Automatic Software Download Is Working Correctly (Junos OS)

IN THIS SECTION

- Purpose | 184
- Action | 184
- Meaning | 185

Purpose

Verify that the automatic software download feature is working correctly.

Action

Use the `show system services dhcp client interface-name` command to verify that the automatic software download feature has been used to install a software package.

```
user@switch> show system services dhcp client ge-0/0/1.0
Logical Interface Name      ge-0/0/1.0
Hardware address            00:0a:12:00:12:12
Client Status               bound
Vendor Identifier           ether
Server Address              10.1.1.1
Address obtained            10.1.1.89
Lease Obtained at           2022-08-20 18:13:04 PST
Lease Expires at           2022-08-22 18:13:04 PST

DHCP Options :
Name: name-server, Value: [ 10.209.194.131, 203.0.113.2, 203.0.113.3 ]
Name: server-identifier, Value: 10.1.1.1
```



```
Name: router, Value: [ 10.1.1.80 ]
Name: boot-image,
Value: junos-install-ex-arm-64-22.2R1.3.tgz
Name: boot-image-location,
Value: 10.1.1.25:/bootfiles/
```

Meaning

The output from this command shows the name and location of the software package under DHCP options when automatic software download was last used to install a software package. The sample output in DHCP options shows that the last DHCP server message to arrive on the DHCP client had a boot server address of 10.1.1.1 and a boot file named **junos-install-ex-arm-64-22.2R1.3.tgz**. If automatic software download was enabled on this client switch during the last DHCP message exchange, these values were used by the switch to upgrade the software.

RELATED DOCUMENTATION

| *Configuring a DHCP Server on Switches*

Installing Junos OS Software with Junos Automation Enhancements

Before you install software, download the Junos OS `jinstall-qfx-5-flex-x.tgz` software bundle. For information on downloading and accessing the files, see "[Installing Software Packages on QFX Series Devices \(Junos OS\)](#)" on page 174.

Junos operating system (Junos OS) with Junos Automation Enhancements is a full-featured version of Junos OS with Veriexec disabled, which can only be installed on supported devices.



NOTE: You must install the `jinstall-qfx-5-flex-x.tgz` software bundle in order to use the automation enhancements.



BEST PRACTICE: Before you install the software, back up any critical files in `/var/home`. For more information regarding how to back up critical files, contact Customer Support at <https://www.juniper.net/support>.

Install the software:

1. Issue the `request system software add` command with the `validate` option:

- If the installation package resides locally on the switch, issue the **request system software add validate *pathname source* reboot** command, using the following format:

```
user@switch> request system software add validate /var/tmp/jinstall-qfx-5-flex-x.tgz reboot
```

- If the installation package resides remotely, issue the **request system software add validate *pathname source* reboot** command, using the following format:

```
user@switch> request system software add validate ftp://ftpserver/directory/jinstall-qfx-5-flex-x.tgz reboot
```

2. After the reboot has finished, verify that the new version of software has been properly installed by executing the **show version** command.

```
user@switch> show version
```

```
fpc0:
```

```
-----
```

```
Hostname: qfx5100-24q-et013
```

```
Model: qfx5100-24q-2p
```

```
JUNOS Base OS Software Suite [13.2X51-D20]
```

```
JUNOS Base OS boot [13.2X51-D20]
```

```
JUNOS Crypto Software Suite [13.2X51-D20]
```

```
JUNOS Online Documentation [13.2X51-D20]
```

```
JUNOS Kernel Software Suite [13.2X51-D20]
```

```
JUNOS Packet Forwarding Engine Support (qfx-x86-32) [13.2X51-D20]
```

```
JUNOS Routing Software Suite [13.2X51-D20]
```

```
JUNOS Enterprise Software Suite [13.2X51-D20]
```

```
JUNOS py-base-i386 [13.2X51-D20]
```

```
Puppet on Junos [2.7.19_1.junos.i386]
```

```
Ruby Interpreter [11.10.4_1.junos.i386]
```

```
Chef [11.10.4_1.junos.i386]
```

```
junos-ez-stdlib [11.10.4_1.junos.i386]
```

```
JUNOS Host Software [13.2X51-D20]
```

```
JUNOS for Automation Enhancement
```



NOTE: If you are upgrading a device from standard Junos OS to use Junos Automation Enhancements and you are *not* loading the new factory default configuration, you need to use the following procedure.

1. Edit your existing Junos OS configuration to include the following configuration statements:

```
[edit]
user@switch# set system extensions providers juniper license-type juniper
deployment-scope commercial
user@switch# set system extensions providers chef license-type juniper deployment-
scope commercial
```



NOTE: The factory default configuration of the QFX5100 switch `jinstall-qfx-5-flex-x.tgz` software bundle is a Layer 3 configuration, whereas the factory default configuration for QFX5100 switch software bundles is a Layer 2 configuration. Therefore, if you are running the `jinstall-qfx-5-flex-x.tgz` software bundle on a QFX5100 switch and you use the `load factory-default` command, the resulting factory default configuration is set up for Layer 3 interfaces.

This is the factory default configuration for QFX5100 switch `jinstall-qfx-5-flex-x.tgz` software bundle:

```
user@switch> show configuration
```

```
system syslog user * any emergency
system syslog file messages any notice
system syslog file messages authorization info
system syslog file interactive-commands interactive-commands any
system extensions providers juniper license-type juniper deployment-scope commercial
system extensions providers chef license-type juniper deployment-scope commercial
system commit factory-settings reset-virtual-chassis-configuration
system commit factory-settings reset-chassis-lcd-menu
system processes app-engine-virtual-machine-management-service traceoptions level notice
system processes app-engine-virtual-machine-management-service traceoptions flag all
interfaces et-0/0/0 unit 0 family inet dhcp vendor-id Juniper-qfx5100-24q-2p
interfaces xe-0/0/0:0 unit 0 family inet dhcp vendor-id Juniper-qfx5100-24q-2p
interfaces xe-0/0/0:1 unit 0 family inet dhcp vendor-id Juniper-qfx5100-24q-2p
```

[illegible]

[illegible]

```
protocols igmp-snooping vlan default
vlans default vlan-id 1
```

SEE ALSO

Overview of Junos Automation Enhancements on Devices Running Junos OS with Enhanced Automation

Installing and Recovering Software Using the Open Network Install Environment (ONIE)

IN THIS SECTION

- [Understanding the Open Network Install Environment | 192](#)
- [Downloading Software Files with a Browser | 193](#)
- [Connecting to the Console Port | 193](#)
- [Backing Up the Current Configuration Files | 194](#)
- [Uninstalling the Existing Version of Junos OS | 194](#)
- [Installing a Junos OS Software Package That Resides on a Webserver or DHCP Server with DHCP Options Configured | 195](#)
- [Installing Junos OS Software Using Secure Copy Protocol \(SCP\) | 196](#)
- [Installing Junos OS Software Using FTP or TFTP Without a Webserver | 197](#)
- [Installing Junos OS Software Using DHCP Server with No DHCP Options Configured | 198](#)
- [Installing Junos OS Software Using Webserver Without DHCP Configured | 199](#)
- [Installing Junos OS Software Using USB Media | 200](#)
- [Verifying Software Installation | 200](#)
- [Troubleshooting Boot Problems | 201](#)
- [Creating an Emergency Boot Device | 202](#)
- [Performing a Recovery Installation | 203](#)

ONIE, the open network install environment from Cumulus Networks, is a network OS installer that installs Junos OS and third party applications on a switch. Juniper Network switches come pre-installed

with ONIE. When you turn on a switch, the ONIE discovery and execution (ODE) application locates the management Ethernet interface and the Junos OS software package, which can be found either locally on the switch or on the network using HTTP, FTP, or TFTP. After the switch discovers and downloads the Junos OS software package, the switch installs the Junos OS software, reboots, and then boots from Junos OS. Junos OS then becomes the default software image.



NOTE: If you want to use the Junos OS CLI to install software, see ["Installing Software Packages on QFX Series Devices \(Junos OS\)" on page 174](#).

Upgrading involves these tasks:

Understanding the Open Network Install Environment

When you log into the switch with ONIE, you see the install boot menu:

- Juniper Linux (This is a default menu option.)
- Juniper Linux Debug
- Juniper Linux Recovery
- Go to ONIE Loader
 - ONIE: Install OS (This is a default menu option.)
 - ONIE: Rescue
 - ONIE: Uninstall OS
 - ONIE: Update ONIE
 - ONIE: Embed ONIE

You can use the following commands to install and uninstall Junos OS and start and stop the ONIE ODE application:

- **onie-nos-install**

Installs Junos OS from any URL, such as http://, ftp://, and file://.

- **onie-uninstaller**

Uninstalls Junos OS.

- **onie-discovery-start**

The discovery process starts automatically. However, if you stop the discovery process by issuing the **onie-discovery-stop** command, you can restart the discovery process by issuing the **onie-discovery-start** command.

- **onie-discovery-stop**

Stops the discovery process. To restart the discovery process, issue the **onie-discovery-start** command.

Downloading Software Files with a Browser

You download the software package from the Juniper Networks Downloads page at <https://support.juniper.net>.



NOTE: To access the download site, you must have a service contract with Juniper Networks and an access account. If you need help obtaining an account, complete the registration form at the Juniper Networks website <https://www.juniper.net/registration/Register.jsp>.

To download a software package:

1. Using a Web browser, navigate to the <https://support.juniper.net>.
2. Either click **View all products>** and select the product you are downloading software for, or type the product name.
3. Find the package you want and click the item in the Downloads column.
A login screen appears.
4. Enter your name and password and press Enter.
5. Read the End User License Agreement, click the **I agree** radio button, and then click **Proceed**.
6. Save the Junos OS software image file to your computer.

The Junos OS software image file name is presented in the *prefix-release-edition-signed.extension* format. For example, the image name for Junos OS Release 15.1X53-D10 on QFX10000 series switch is **jinstall-qfx-10-f-15.1X53-D10.7-domestic-signed**.

See "[Junos OS Installation Package Names](#)" on page 15 for additional information on image file naming.

7. Open or save the installation package either to the local system in the **var/tmp** directory or to a remote location. If you are copying the installation package to a remote system, make sure that you can access it using HTTP, TFTP, FTP, or SCP.

Connecting to the Console Port

We recommend that you connect to the console port while installing the installation package so you can respond to any required user input and detect any errors that might occur.

Backing Up the Current Configuration Files

Before you install the new installation package, we strongly recommend that you back up your current configuration files because the upgrade process removes all of the stored files on the switch.

To back up your current configuration files, enter the `save` command:

```
user@switch# save filename
```

Executing this command saves a copy of your configuration files to a remote location such as an external USB device.

Uninstalling the Existing Version of Junos OS

The switch comes preinstalled with a version of Junos OS that is to be used with the Junos OS CLI. However, if you want to use ONIE to install Junos OS, you need to uninstall the existing Junos OS and reinstall the Junos OS image that has a .bin extension—for example, `jnpr-qfx-5e-jdm-onie-updater-15.1-20150819_ups.4.bin` file.

To uninstall your existing Junos OS version:

1. Select **Go to ONIE Loader** from the GNU GRUB menu.
2. Select **ONIE: Uninstall OS** from the GNU GRUB menu.

The Junos OS is uninstalled, and the switch reboots.

By default, the ONIE discovery and execution (ODE) application attempts to discover and fetch an image from a configured DHCP or webserver and the management IP address of the switch and the IP address of the default gateway. If you want to manually configure static addressing for the management IP address of the switch, issue `onie-discovery-stop` command at the ONIE prompt, and then manually configure the management IP address and IP address of the default gateway.

For example:

```
ONIE:/ # onie-discovery-stop
ONIE:/ # ifconfig eth0 10.204.32.96 netmask 255.255.254.0

ONIE:/ # route add default gw 10.204.47.254
```

To restart the ONIE discovery and execution (ODE) application, issue the `onie-discovery-start` command.

For example:

```
ONIE:/ # onie-discovery-start
```

Installing a Junos OS Software Package That Resides on a Webserver or DHCP Server with DHCP Options Configured

To install a Junos OS software package residing on a webserver or DHCP server:

1. Copy the software image with the filename `onie-installer` to the `var/www/html` directory of the webserver or DHCP server.
2. Configure the DHCP option 114 in the DHCP server to redirect to the webserver to fetch the Junos OS software image.
3. Uninstall the preinstalled Junos OS version.
 - Select **Go to ONIE Loader** from the GNU GRUB menu.
 - Select **ONIE: Uninstall OS** from the GNU GRUB menu.

The Junos OS is uninstalled, and the switch reboots.

4. Configure DHCP option 114 and other DHCP options as necessary.

Here is a sample Windows Open DHCP server configuration with DHCP option 114 configured.

```
#Following are range-specific DHCP options.
#You can copy more option names from [GLOBAL_OPTIONS]
IP=10.204.42.250
SubnetMask=255.255.240.0
Router=10.204.47.254
114="http://10.207.66.147/onie-installer"
```

Here is a sample boot initialization log, showing the options you just configured:

```
Info: Trying DHCPv4 on interface: eth0
ONIE: Using DHCPv4 addr: eth0: 10.204.42.250 / 255.255.240.0
ONIE: Starting ONIE Service Discovery
Info: Fetching http://10.207.66.147/onie-installer ...
ONIE: Executing installer: http://10.207.66.147/onie-installer <----- automatically
redirects to web sever to fetch Junos OS image.
Verifying image checksum ... OK.
```

```
Preparing image archive ... OK.
Installing Juniper NOS...
```

The log shows that the installation process has fetched the Junos OS software image from the DHCP server and is installing the Junos OS software.

The switch reboots and the GNU GRUB menu is displayed.

Installing Junos OS Software Using Secure Copy Protocol (SCP)

To install Junos OS software using SCP:

1. Uninstall the preinstalled Junos OS version.
 - Select **Go to ONIE Loader** from the GNU GRUB menu.
 - Select **ONIE: Uninstall OS** from the GNU GRUB menu.

The Junos OS is uninstalled, and the switch reboots.

By default, the ONIE discovery and execution (ODE) application attempts to discover and fetch an image from a configured webserver. If you do not have DHCP configured, you will need to stop the ONIE discovery and execution (ODE) application and manually configure static addressing for the management IP address of the switch,

For example:

```
ONIE:/ # onie-discovery-stop
ONIE:/ # ifconfig eth0 10.204.32.96 netmask 255.255.254.0

ONIE:/ # route add default gw 10.204.47.254
```

2. Use SCP to copy the Junos OS image from a server or other location to the **/var/tmp** directory on the switch.

For example:

```
user@server scp jnpr-qfx-5e-jdm-onie-updater-15.1-20150819_ups.4.bin
root@10.204.32.196:/var/tmp/
```

3. Issue the **onie-nos-install** command in the **/var/tmp** directory to install Junos OS software.

```
ONIE:/var/tmp # onie-nos-install file:///var/tmp/jnpr-qfx-5e-jdm-onie-
updater-15.1-20150819_ups.4.bin
```

The switch reboots and displays the GNU GRUB menu.

Installing Junos OS Software Using FTP or TFTP Without a Webserver

To install Junos OS software using FTP or TFTP:

1. Uninstall the preinstalled Junos OS version.
 - Select **Go to ONIE Loader** from the GNU GRUB menu.
 - Select **ONIE: Uninstall OS** from the GNU GRUB menu.

The Junos OS is uninstalled, and the switch reboots.

By default, the ONIE discovery and execution (ODE) application attempts to discover and fetch an image from a configured webserver. If you do not have DHCP configured, you will need to stop the ONIE discovery and execution (ODE) application and manually configure static addressing for the management IP address of the switch,

For example:

```
ONIE:/ # onie-discovery-stop
ONIE:/ # ifconfig eth0 10.204.32.96 netmask 255.255.254.0

ONIE:/ # route add default gw 10.204.47.254
```

2. Copy the Junos OS image to an FTP or TFTP directory.
3. Issue the **onie-nos-install** command at the ONIE prompt to install the Junos OS software.

If you are using FTP:

```
ONIE:/ # onie-nos-install ftp://<username>:<password>@10.209.152.22/jnpr-qfx-5e-jdm-onie-
updater-15.1-20150819_ups.4.bin
```

If you are using TFTP:



NOTE: The software image should be located in the **/tftp/boot** directory.

```
ONIE:/ # onie-nos-install tftp://10.207.66.147/jnpr-qfx-5e-jdm-onie-
updater-15.1-20150819_ups.4.bin
```

The switch reboots and displays the GNU GRUB menu.

Installing Junos OS Software Using DHCP Server with No DHCP Options Configured

Use this installation method if you cannot modify or set the DHCP options on your DHCP server.

To install the Junos OS software using a DHCP server with no DHCP options configured:

1. Copy the software image with the filename `jnpr-qfx-5e-jdm-onie-updater-15.1-20150819_ups.4.bin` to the `var/www/html` directory of the webserver or DHCP server.
2. Uninstall the preinstalled Junos OS version.
 - Select **Go to ONIE Loader** from the GNU GRUB menu.
 - Select **ONIE: Uninstall OS** from the GNU GRUB menu.

The Junos OS is uninstalled, and the switch reboots.

3. Issue the `onie-nos-install` command at the ONIE prompt to install the Junos OS software.

For example:

```
ONIE:/ # onie-nos-install http://10.207.66.147/jnpr-qfx-5e-jdm-onie-
updater-15.1-20150819_ups.4.bin
```

Here is sample log with the options you just configured:

```
ONIE:/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 94:DE:80:AA:F2:E1
          inet addr:10.204.42.250  Bcast:10.204.47.255  Mask:255.255.240.0  <<<---- -->
          Received IP address from DHCP server, but auto redirected to web server. Installation will
          not happen because DHCP option (114) is not configured.

          inet6 addr: fe80::96de:80ff:feaa:f2e1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:444 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:48170 (47.0 KiB)  TX bytes:2678 (2.6 KiB)
          Memory:80180000-801a0000
```

This log shows that the installation process has fetched the Junos OS software image from the webserver and is installing the Junos OS software.

```
Stopping: discover... done.
Info: Fetching http://10.207.66.147/jinstall-qfx-10-f-15.1X53-D10.7-domestic-signed.tgz  ...
```

```

Connecting to 10.207.66.147 (10.207.66.147:80)
installer          100% |*****| 464M 0:00:00 ETA
ONIE: Executing installer: http://10.207.66.147/jnpr-qfx-5e-jdm-onie-
updater-15.1-20150819_ups.4.bin
Verifying image checksum ... OK.
Preparing image archive ... OK.
Installing Juniper NOS...

```

The switch reboots and the GNU GRUB menu is displayed.

Installing Junos OS Software Using Webserver Without DHCP Configured

Use this installation method if you do not have a DHCP server.

To install the Junos OS software using a webserver without DHCP configured:

1. Because the switch comes preinstalled with the Junos OS to be used with the Junos OS CLI, you need to uninstall this version of software before you can install the Junos OS image to be used with ONIE.
 - Select **Go to ONIE Loader** from the GNU GRUB menu.
 - Select **ONIE: Uninstall OS** from the GNU GRUB menu.

The Junos OS is uninstalled, and the switch reboots.

2. Select **ONIE: Uninstall OS** from the GNU GRUB menu.

The Junos OS is uninstalled, and the switch reboots.

By default, the ONIE discovery and execution (ODE) application attempts to discover and fetch an image from a configured webserver. Because you do not have DHCP configured, you will need to stop the ONIE discovery and execution (ODE) application and manually configure static addressing for the management IP address of the switch.

For example:

```

ONIE:/ # onie-discovery-stop
ONIE:/ # ifconfig eth0 10.204.32.96 netmask 255.255.254.0

ONIE:/ # route add default gw 10.204.47.254

```

3. Copy the software image to the **var/www/html** directory of the webserver.
4. Issue the **onie-nos-install** command at the ONIE prompt to install the Junos OS software.

For example:

```
ONIE:/ # onie-nos-install http://10.204.35.100/jnpr-qfx-5e-jdm-onie-
updater-15.1-20150819_ups.4.bin
```

Here is sample log:

```
Stopping: discover... done.
Info: Fetching http://10.204.35.100/jnpr-qfx-5e-jdm-onie-updater-15.1-20150819_ups.4.bin ...
Connecting to 10.204.35.100 (10.204.35.100:80)
installer          100% |*****| 464M 0:00:00 ETA
ONIE: Executing installer: http://10.204.35.100/jnpr-qfx-5e-jdm-onie-
updater-15.1-20150819_ups.4.bin
Verifying image checksum ... OK.
Preparing image archive ... OK.
Installing Juniper NOS...
```

The log shows that the installation process has fetched the Junos OS software image from the webserver and is installing the Junos OS software.

The switch reboots and the GNU GRUB menu is displayed.

Installing Junos OS Software Using USB Media

ONIE installation from a Junos OS image stored on USB media is not currently supported.

Use another procedure from this document to install ONIE.

Verifying Software Installation

IN THIS SECTION

- Purpose | 200
- Action | 201

Purpose

Verify that the software was installed successfully on the switch.

Action

To verify that the software was properly installed, issue the `show version` command.

```
user@switch > show version
```

Troubleshooting Boot Problems

IN THIS SECTION

- [Problem | 201](#)
- [Solution | 201](#)

Problem

Description

Junos OS does not boot.

Solution

If Junos OS does not boot, and the console displays the Yocto GNU Linux shell instead, it could mean that you have booted in the Juniper Linux Debug mode. If you see an error message that says, “[**Error**] **Does not seem to be an QFX10002.**” could mean that the EEPROM does not contain vendor-specific information. To verify the vendor-specific information, perform an ONIE: Rescue installation, and then verify the contents of the `/var/run/*.dat` file.

1. Select **ONIE: Rescue** from the GNU GRUB menu.
2. Issue the `onie-syseeprom` at the ONIE prompt.

For example:

```
ONIE:/ # onie-syseeprom
TlvInfo Header:
  Id String:    TlvInfo
  Version:      1
  Total Length: 315
```

TLV Name	Code	Len	Value
Base MAC Address	0x24	6	54:2A:A2:FB:DC:00
MAC Addresses	0x2A	2	256
Product Name	0x21	23	QFX10000-ÿÿÿÿÿÿÿ
Serial Number	0x23	12	116G1EC00032
Part Number	0x22	16	1AES48S6Q.A2Gÿÿÿ
Device Version	0x26	1	1
Manufacture Date	0x25	19	01/13/2015 21:40:30
Vendor Name	0x2D	20	JUNIPER NETWORKS INC
Manufacturer	0x2B	14	JUNIPER NETWORKS INC
Vendor Extension	0xFD	48	0x00 0x00 0x7C 0x82 0x01 0x00 0x41 0x32 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x0F
Vendor Extension	0xFD	62	0x00 0x00 0x0A 0x4C 0x51 0x06 0x52 0x45 0x56 0x20 0x30 0x31 0x52 0x0C 0x3F
Platform Name	0x28	37	x86_64-alpha_networks_snx60a0_486f-r0
Loader Version	0x29	23	master-201412161452.0.1
CRC-32	0xFE	4	0xB88C8885

Checksum is valid.

From the output, you can see that the vendor-specific information confirms that it is for Juniper Networks.

Creating an Emergency Boot Device

Before you begin, you need to have the `jnpr-qfx-5e-jdm-onie-updater-15.1-20150819_ups.4.bin` version of ONIE software.

If the Open Network Install Environment (ONIE) software is damaged or corrupted in some way, or the switch went into rescue mode, you can use an emergency boot device to repartition the primary disk and load a fresh installation of ONIE. Use the following procedure to create an emergency boot device.



NOTE: In the following procedure, we assume that you are creating the emergency boot device on a switch. You can create the emergency boot device on any PC or laptop that supports Linux.

To create an emergency boot device:

1. Insert the USB device into the front USB port of the switch.

Make sure the USB device is at least 1GB.

2. Issue the following command from the directory on the switch in which the ISO file is located:

```
ONIE:/ # jnpr-qfx-5e-jdm-onie-updater-15.1-20150819_ups.4.bin of=<usb-detected-drive> bs=1M
```

You can also issue the **dd** command using the full path to where the ISO file is located.

For example, if the ISO file is located in the **/var/tmp/** directory:

```
ONIE:/ # dd if=/var/tmp/jnpr-qfx-5e-jdm-onie-updater-15.1-20150819_ups.4.bin of=<usb-detected-drive> bs=1M
```

The switch writes the installation media image to the USB device:

3. Remove the USB device from the USB port of the switch.

Performing a Recovery Installation

In the event that the Open Network Install Environment (ONIE) is corrupted, the switch goes into rescue mode, or you need to reinstall ONIE software for any reason, you need to perform a recovery installation.



NOTE: All Junos OS partitions are destroyed during a recovery installation.



NOTE: Before you can perform a recovery installation, make sure you have an emergency boot device loaded with ONIE software.

1. Insert the emergency boot device into the device.
2. Power cycle the device.
3. Press the **ESC** button to go into the Boot Manager menu.
4. Select **Boot Manager**, and then press **Enter**.
5. Select **Unigen PQS1000** under **Legacy USB**, and then press **Enter**.
6. Select **ONIE: Embed ONIE** from the **ONIE Installer** menu, and then press **Enter**.

The recovery installation proceeds using the emergency boot device.

7. Remove the emergency boot device.
8. Verify that the ONIE software was installed by looking at the installation log file.

For example:

```
Info: Found static url: file:///lib/onie/onie-updater
ONIE: Executing installer: file:///lib/onie/onie-updater
Verifying image checksum ... OK.
Preparing image archive ... OK.
ONIE: Version      : master-201412161452.0.1
```

Installation log files are displayed automatically during the installation process, but if you want to verify installation log files at a different time, you can find them in the in the `/var/log/` directory. To view an installation log file, issue the `tail -f /var/log/onie.log` command.

9. Issue the `parted /dev/sda print` command to verify that the ONIE partitions have been created.

For example:

```
ONIE:/ # parted /dev/sda print
Model: ATA TS8GHSD630 (scsi)
Disk /dev/sda: 8012MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	3146kB	2097kB		GRUB-BOOT	hidden, bios_grub
2	3146kB	137MB	134MB	ext4	ONIE-BOOT	hidden

RELATED DOCUMENTATION

Installing Software Packages on QFX Series Devices (Junos OS) 174
Upgrading Software by Using Automatic Software Download for Switches (Junos OS) 182
<i>DHCP Server Configuration</i>

Manage YANG Packages During an Upgrade or Downgrade

IN THIS CHAPTER

- [Managing YANG Packages and Configurations During a Software Upgrade or Downgrade | 205](#)

Managing YANG Packages and Configurations During a Software Upgrade or Downgrade

IN THIS SECTION

- [Backing up and Deleting the Configuration Data | 206](#)
- [Restoring the YANG Packages and Configuration Data | 207](#)

Certain devices running Junos OS enable you to load custom YANG modules on the device to add data models that are not natively supported by Junos OS. When you add, update, or delete a YANG data model, Junos OS rebuilds its schema and then validates the active configuration against the updated schema.

When you upgrade or downgrade Junos OS, by default, the system validates the software package or bundle against the current configuration. During the installation, the schema for custom YANG data models is not available. As a result, if the active configuration contains dependencies on these models, the software validation fails, which causes the upgrade or downgrade to fail.

In addition, devices that are running Junos OS based on FreeBSD version 6 remove custom YANG packages from the device during the software installation process. For this Junos OS variant, if the active configuration contains dependencies on custom YANG data models, the software installation fails even if you do not validate the software against the configuration, because the configuration data cannot be validated during the initial boot-time commit.

For these reasons, before you upgrade or downgrade the Junos OS image on a device that has one or more custom YANG modules added to it, you must remove all configuration data corresponding to the custom YANG data models from the active configuration. After the software installation is complete, add the YANG packages and corresponding configuration data back to the device, if appropriate. The tasks are outlined in this topic.



NOTE: You do not need to delete configuration data corresponding to OpenConfig packages before upgrading or downgrading Junos OS.

Backing up and Deleting the Configuration Data

If the configuration contains dependencies on custom YANG data models:

1. If you plan to restore the configuration data that corresponds to the nonnative YANG data models after the software is updated, save a copy of either the entire configuration or the configuration data corresponding to the YANG data models, as appropriate.

- To save the entire configuration:

```
user@host> show configuration | save (filename | url)
```

- To save configuration data under a specific hierarchy level:

```
user@host> show configuration path-to-yang-statement-hierarchy | save (filename | url)
```

2. In configuration mode, delete the portions of the configuration that depend on the custom YANG data models.

```
[edit]
user@host# delete path-to-yang-statement-hierarchy
```

3. Commit the changes.

```
[edit]
user@host# commit
```

4. Prior to performing the software installation, ensure that the saved configuration data and the YANG module and script files are saved to a local or remote location that will preserve the files during the installation and that will be accessible after the installation is complete.

Restoring the YANG Packages and Configuration Data

After the software installation is complete, load the YANG packages onto the device (where required), and restore the configuration data associated with the packages, if appropriate. During a software upgrade or downgrade, devices running Junos OS with upgraded FreeBSD preserve custom YANG packages, whereas devices running Junos OS based on FreeBSD version 6 delete the packages.

1. Load the YANG packages (devices running Junos OS based on FreeBSD version 6 only).

```
user@host> request system yang add package package-name module [modules] deviation-  
module [modules] translation-script [scripts] action-script [scripts]
```

2. When the system prompts you to restart the Junos OS CLI, press Enter to accept the default value of yes.

```
...  
WARNING: cli has been replaced by an updated version:  
...  
Restart cli using the new version ? [yes,no] (yes)  
  
Restarting cli ...
```



NOTE: To prevent CLI-related or configuration database errors, we recommend that you do not perform any CLI operations, change the configuration, or terminate the operation while a device is in the process of adding, updating, or deleting a YANG package and modifying the schema.

3. In configuration mode, load the configuration data associated with the YANG packages.

For example, to load the configuration data from a file relative to the top level of the configuration statement hierarchy:

```
[edit]  
user@host# load merge (filename | url)
```



NOTE: For more information about loading configuration data, see the *CLI User Guide*.

4. Commit the changes.

```
[edit]  
user@host# commit
```

RELATED DOCUMENTATION

| *Managing YANG Packages, Modules, and Scripts on Devices Running Junos OS*

3

PART

Install Software on Firewalls

- Junos OS Upgrades for SRX Series Firewalls | **210**
 - Prepare the USB Flash Drive to Upgrade Junos OS on SRX300 Line of Firewall Devices | **231**
 - Upgrade and Install the Boot Loader on SRX Series Firewalls | **237**
 - Configure Root Partitions on SRX Series Devices | **254**
 - Veriexec and OpenPGP | **262**
-

Junos OS Upgrades for SRX Series Firewalls

IN THIS CHAPTER

- [Upgrades for SRX Series Firewalls | 210](#)
- [Upgrade Junos OS Package on SRX Series Firewalls | 211](#)
- [Install Junos OS on SRX Series Firewalls Using the Partition Option | 216](#)
- [Revert the Junos OS Software Image to the Previous Release | 220](#)
- [Reboot and Halt SRX Series Firewall | 223](#)
- [Manage Chassis Component Status on SRX Series Firewalls: Online and Offline Operations | 229](#)

Upgrades for SRX Series Firewalls

SUMMARY

Learn how to upgrade Junos OS on SRX Series Firewalls with Juniper Networks, covering preparation, software download, verification, execution, and troubleshooting for a smooth transition and minimal downtime.

IN THIS SECTION

- [Junos OS Upgrade Methods on SRX Series Firewalls | 211](#)

SRX Series Firewalls are delivered with Junos OS pre-installed on them. When you power on a device, it starts (boots) up using its primary boot device. These devices also support secondary boot devices, enabling you to back up your primary boot device and configuration.

On an SRX Series Firewalls, you can configure the primary and secondary boot device each with a snapshot of the current configuration, the default factory configuration, or a rescue configuration. You can also replicate the configuration from one device to the other device on a firewall.

Your primary boot device may get corrupted and your firewall device may not have a secondary boot device configured. To resolve this issue, reload the Junos OS package onto the corrupted storage device from a USB flash drive or a Trivial File Transfer Protocol (TFTP) server.

As new features and software fixes become available, you must upgrade Junos OS to use them. Before an upgrade, ensure that you back up your primary boot device.

Junos OS Upgrade Methods on SRX Series Firewalls

Certain Junos OS upgrade methods format the device before installation, whereas other methods do not. To install Junos OS with dual-root partitioning, you must use an upgrade method that formats the device before installation.

Table 16: Junos OS Upgrade Methods and Media-Formatting Behavior

Upgrade Methods	Media-Formatting Behavior
Installation from the boot loader using a TFTP server	Formats device before installation
Installation from the boot loader using a USB storage device	Formats device before installation
Installation using the J-Web UI	<ul style="list-style-type: none"> • Formats device before installation • Retains existing partitioning
Installation using the CLI	Retains existing partitioning

After an upgrade that formats the device with dual-root partitioning, you can use the conventional CLI or J-Web UI installation methods for subsequent upgrades.

Upgrade Junos OS Package on SRX Series Firewalls

SUMMARY

Learn how to install Junos OS upgrade packages on SRX Series Firewalls from Juniper Networks. This topic covers the necessary steps to prepare for the upgrade, execute the installation, and verify the successful update of the Junos OS on your firewall devices.

IN THIS SECTION

- [Requirements | 212](#)
- [Installation Overview | 213](#)
- [Configuration | 213](#)
- [Verify Junos OS Upgrade | 215](#)

Requirements

IN THIS SECTION

- [Know the Upgrade Path for Junos OS | 212](#)

Before you begin:

- Verify the available space on the device.
- Download the software package. See [Downloads](#) to download the software package for your products.
- We recommend that you copy the software package to the `/var/tmp` directory from any local directory on the device where you have saved the package after the download. To copy the software package to the `/var/tmp` directory, use the following command from the operational

```
user@host> file copy /var/tmp/install/image-name/var/tmp/
```

Example:

```
user@host> file copy /var/tmp/install/junos-srxsme-10.0R2-domestic.tgz /var/tmp/
```

Know the Upgrade Path for Junos OS

Knowing the upgrade path helps you to choose the correct Junos OS package or packages to install.

Topic	Details
Suggested Releases to Consider	Always check the most up-to-date version as recommended in the Knowledge Base article: See Junos Software Versions - Suggested Releases to Consider and Evaluate .
Upgrade Path Reference	For information about upgrade path, see Junos Upgrade Paths for SRX Platforms .

(Continued)

Topic	Details
Minimal Downtime Procedure	For SRX300 line of Firewalls, if you are upgrading to Junos OS Release 24.4R1 or later, you must use either the procedures outlined in KB 85650 or the minimal downtime procedure documented in KB17947 (Minimal_Downtime_Upgrade_Branch_Mid PDF file) .
ISSU Limitations	Check article on ISSU limitations. See SRX ISSU ICU Upgrade Limitations .

Installation Overview

When you install software on a Junos OS storage device, the system validates the package by default. The request system software add *package-name* command automatically uses the validate option to check the package against the current configuration. This action ensures the device can reboot successfully after installation.

Starting with Junos OS Release 24.4R1, Juniper has renamed software images as:

- junos-install-srxsme* for Release 24.4R1 and later.
- junos-srxsme* for Release 23.4R1 and earlier.

Configuration

IN THIS SECTION

Procedure | 213

Procedure

GUI Quick Configuration

Step-By-Step Procedure

To install Junos OS upgrades on SRX Series Firewalls:

1. In the J-Web UI, select **Device Administration>Software Management>Install Package**.
2. On the Upload Package page, click **Browse** to navigate to the software package location and select the package, for example, **junos-srxsme-24.4R1-domestic.tgz**.
3. Select the **Reboot If Required** check box to set the device to reboot automatically when the upgrade is complete.
4. Select the **Do not save backup** check box to bypass saving the backup copy of the current Junos OS package.
5. Click **Upload Package**. The software is activated after the device reboots.
6. Click **OK** to save your changes as a candidate configuration. In Junos OS, a candidate configuration is a draft, that is, changes are saved but not yet applied to the running system.
7. Click **Commit Options>Commit**. Commit applies the candidate configuration to the device and activates all changes.

Step-by-Step Procedure Using CLI

Follow these steps to install a Junos OS software package using the CLI.

Before You Begin:

If you're new to Junos OS, see *Using the CLI Editor in Configuration Mode* to understand how to navigate the configuration hierarchy.

1. From **operational mode**, install the software using the `request system software add` command with the required options:

```
user@host> request system software add /var/tmp/junos-srxsme-24.4R1-domestic.tgz no-copy
```

- Use the `no-copy` option to install the software without saving a copy locally.
- Use the `no-validate` option only if advised by Juniper Networks Technical Assistance Center (JTAC).
- `reboot` – Reboots the device automatically after installation is complete.

2. After the installation is complete, reboot the device to activate the new software:

```
user@host> request system reboot
```

- When the reboot is complete, the device displays the login prompt.

3. After logging in, enter **configuration mode** and verify your settings. If the output doesn't show the intended configuration, repeat the upgrade steps as needed.
4. From configuration mode, confirm your upgrade by entering the `show version` command.

If the output confirms a successful upgrade, enter `commit` from configuration mode.

Verify Junos OS Upgrade

Purpose

Verify that the Junos OS upgrade package is installed.

Action

From operational mode, enter the `show version` command.

Sample Output

`show version`

```
user@host> show version
Hostname: srx345-a
Model: srx345
Junos: 24.4R1-S3.2
JUNOS Software Release [24.4R1-S3.2]
```

Meaning

The `show version` command displays the hostname, model number, and the release information loaded on the device.

Install Junos OS on SRX Series Firewalls Using the Partition Option

SUMMARY

Learn how to install Junos OS with the partition option which helps format the internal media, installs the image with dual-root partitioning, and reboots the device to complete the setup.

IN THIS SECTION

- [Requirements | 216](#)
- [Installation Overview | 216](#)
- [Configuration | 217](#)
- [Verify the Partition Scheme Details | 219](#)

Requirements

Before you start the installation, back up important system data and configurations. This step ensures that you can quickly recover your device in case of an upgrade failure.

Installation Overview

To reinstall Junos OS with dual-root partitioning, use the `request system software add` command with the `partition` and `reboot` options.

The `request system software add` command:

- Formats the internal media.
- Installs the image with dual-root partitioning.
- Reboots the device to complete the setup. The device also boots up with the image installed with the dual-root partitioning scheme.

When you use the `partition` option, the internal media formatting and image installation are scheduled to run on the next reboot. Therefore, we recommend that you use this option with the `reboot` option.

The installation process takes 15–20 minutes, during which, the device is unreachable. The process erases all data except the current configuration, so back up any important files beforehand.

In Junos OS Release 24.4R1, we renamed the software install images.

The system recognizes images named:

- `junos-installrsrxme*` to install Junos OS Release 24.4R1 or later releases.
- `junossrxsme*` to install Junos OS Release 23.4R1 or previous releases.

In this example, add the software package `junos-srxsme-10.0R2-domestic.tgz` with the following options:

Option	Purpose
<code>no-copy</code>	Installs without saving a local copy (saves space)
<code>no-validate</code>	Skips configuration validation (use only if advised by JTAC)
<code>partition</code>	Formats and sets up dual-root partitioning
<code>reboot</code>	Reboots the device after installation

Configuration

IN THIS SECTION

- [Procedure | 217](#)

Procedure

CLI Quick Configuration

To install Junos OS with the `partition` option, enter the following command from operational mode:

```
user@host>request system software add junos-srxsme-10.0R2-domestic.tgz no-copy no-validate  
partition reboot
```

Step-By-Step Procedure

To install Junos OS with the `partition` option:

1. In the J-Web UI, select **Device Administration>Software Management>Install Package**.
2. On the Install Package page, specify the FTP or HTTP server, file path, and software package name. Type the full address of the software package location on the FTP or HTTP. Example: **ftp://**

hostname/pathname/junos-srxsme-xx.0R2-domestic.tgz or <http://hostname/pathname/junos-srxsme-xx.0R2-domestic.tgz>.

Specify the username and password, if the server requires one.

3. Click the **Reboot If Required** check box to set the device to reboot automatically when the upgrade is complete.
4. Click the **Do not save backup** check box to bypass saving the backup copy of the current Junos OS package.
5. Click the **Format and re-partition the media before installation** check box to format the internal media with dual-root partitioning.
6. Click **Fetch and Install Package**. The software is activated after the device reboots.

This formats the internal media and installs the new Junos OS image on the media with dual-root partitioning.

Step-by-Step Procedure Using CLI

The following example requires you to navigate various levels in the configuration hierarchy. For detailed instructions, see *Using the CLI Editor in Configuration Mode*.

To install Junos OS Release 10.0 or later with the partition option:

1. Upgrade the device to Junos OS using the CLI.
2. After the device reboots, upgrade the boot loader to the latest version. See *Preparing the USB Flash Drive to Upgrade Junos OS on SRX Series Devices*.
3. Reinstall the Junos OS Release 10.0 or later image.

```
user@host>request system software add junos-srxsme-10.0R2-domestic.tgz no-copy no-validate
partition reboot
Copying package junos-srxsme-10.0R2-domestic.tgz to var/tmp/install
Rebooting ...
```

4. From configuration mode, confirm your configuration by entering the `show system storage partitions` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. If the output shows the intended configuration, enter `commit` from configuration mode.

Sample Output

Sample output on a system with dual-root partitioning:

```
user@host> show system storage partitions
```

```
Boot Media: internal (da0)
Active Partition: da0s2a
Backup Partition: da0s1a
Currently booted from: active (da0s2a)
```

Partitions Information:

Partition	Size	Mountpoint
s1a	293M	altroot
s2a	293M	/
s3e	24M	/config
s3f	342M	/var
s4a	30M	recovery

If the output shows the intended configuration, enter `commit` from the configuration mode.

Verify the Partition Scheme Details

Confirm that the configuration is working properly.

Purpose

Verify that the partitioning scheme on the SRX Series Firewall are configured.

Action

From operational mode, enter the `show system storage partitions` command.

Meaning

Verifies whether partitioning scheme details on the SRX Series Firewall are configured correctly.

RELATED DOCUMENTATION

| [Configure Root Partitions on SRX Series Devices](#) | 254

Revert the Junos OS Software Image to the Previous Release

SUMMARY

Learn how to revert the Junos OS software image on the SRX Series Firewalls.

IN THIS SECTION

- [Requirements | 220](#)
- [Overview | 220](#)
- [Configuration | 221](#)
- [Verify Installation of Junos OS Downgrade Package | 222](#)
- [Platform-Specific Rollback Behavior | 222](#)

Requirements

No special configuration beyond device initialization is required before this task.

Overview

When you upgrade your software, the device creates a backup image of the software that was previously installed in addition to installing the requested software upgrade package.

To downgrade the software, you can revert to the previous image by using the backup image. You can use this method to downgrade to only the software release that was installed on the device before the current release. To downgrade to an earlier release, follow the procedure for upgrading, using the software image labeled with the appropriate release. This example returns the software to the previous Junos OS release.

This procedure applies only to downgrading from one Junos OS software release to another or from one Junos OS services release to another.

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Review the "[Platform-Specific Rollback Behavior](#)" on [page 222](#) section for notes related to your platform.

Configuration

IN THIS SECTION

- [Procedure](#) | 221

Procedure

CLI Quick Configuration

To quickly set up this section, copy the following commands into a text file. Remove any line breaks from the commands and update the details to match your network. Paste the commands into the CLI at the [edit] hierarchy level and enter `commit` to apply the changes.

From operational mode, enter:

```
user@host> request system software rollback
request system reboot
```

Step-By-Step Procedure

To downgrade Junos OS on SRX Series Firewalls:

1. In the J-Web UI, select **Device Administration>Software Management>Downgrade**. The image of the previous release (if any) appears on the UI.
2. Click **Downgrade** to downgrade to the previous version of the software or click **Cancel** to cancel the downgrade process.
3. Click **Device Administration>Software Management>Reboot** from the J-Web UI to reboot the device. To downgrade to an earlier release, follow the procedure used to upgrade the software image labeled with the appropriate release.
4. Click **OK** to check your configuration and save your changes as a candidate configuration. In Junos OS, a candidate configuration is a draft, that is, changes are saved but not yet applied to the running system.
5. Click **Commit Options>Commit**. Commit applies the candidate configuration to the device and activates all changes.

Step-by-Step Procedure Using CLI

The following example requires you to navigate various levels in the configuration hierarchy. For detailed instructions, see *Using the CLI Editor in Configuration Mode*.

To downgrade Junos OS on SRX Series Firewalls:

1. From operational mode, use this command to return to the previous Junos OS release.

```
user@host> request system software rollback
```

2. Reboot the device.

```
user@host> request system reboot
```

The device now runs the previous version of Junos OS.

Verify Installation of Junos OS Downgrade Package

Confirm that the configuration is working properly.

Purpose

Verify that the Junos OS downgrade is installed.

Action

From configuration mode, confirm software downgrade by entering the `show system` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. If you are done downgrading the software release, enter `commit` from configuration mode.

correct it.

Meaning

The command `show system` helps verify whether the Junos OS downgrade was installed correctly.

Platform-Specific Rollback Behavior

Use [Feature Explorer](#) to confirm platform and release support for specific features.

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

Platform	Difference
SRX Series	<ul style="list-style-type: none"> • For SRX300 line of Firewalls, you must treat the rollback as a downgrade. • To roll back from Junos OS Release 24.4R1 to either Junos OS Release 23.4R2-S3 or 24.2R2, use the request system software add package-name no-validate reboot command. Avoid using the request system rollback command to roll back to a previous release. • On the SRX300, SRX320, SRX340 and SRX345 Firewalls, the downgrade process sometimes fails to complete because the /oam partition lacks sufficient free space.

Reboot and Halt SRX Series Firewall

SUMMARY

Learn how to reboot and halt SRX Series Firewalls through the CLI and J-Web.

IN THIS SECTION

- [Reboot SRX Series Firewall | 223](#)
- [Halt the SRX Series Firewalls | 226](#)
- [Platform-Specific Halting Software Processes Behavior | 228](#)

Reboot SRX Series Firewall

IN THIS SECTION

- [Requirements | 224](#)
- [Overview | 224](#)
- [Configuration | 224](#)

- [Verify the Firewall Reboot | 225](#)

Requirements

Before rebooting the firewall, save and commit any Junos OS updates.

Overview

Use this example to learn how to reboot a firewall 50 minutes from when you set the time from the internal media.

Configuration

IN THIS SECTION

- [Procedure | 224](#)

Procedure

CLI Quick Configuration

To configure this section quickly, follow these steps:

1. Copy the commands and paste them into a text file.
2. Remove any line breaks and adjust details to match your network.
3. Copy and paste the commands into the CLI at the [edit] level.
4. Enter commit from configuration mode.

From operational mode, enter:

```
user@host> request system reboot at 5 in 50 media internal message stop
```


Step-By-Step Procedure

To reboot an SRX Series Firewall:

1. In the J-Web UI, select **Device Administration>Software Management>Reboot**.
2. Select **Reboot in 50 minutes** to reboot the device 50 minutes from the current time.
3. Click **internal** to select the firewall boot device from the Reboot From Media list.
4. In the Message box, type **stop** as the message to display to any user on the device before the reboot occurs.
5. Click **Schedule**. The J-Web UI requests confirmation to perform the reboot.
6. Click **OK** to confirm the operation.
 - If the reboot is scheduled to occur immediately, the device reboots. You cannot access J-Web until the device restarts and the boot sequence is complete. After the reboot is complete, refresh the browser window to see the J-Web login page.
 - If the reboot is scheduled to occur later, the Reboot page displays the time until reboot. You can optionally cancel the request by clicking **Cancel Reboot** on the Reboot page in J-Web UI.
7. Click **OK** to save your changes as a candidate configuration.
8. If the reboot is successful, click **Commit Options>Commit**.

Results

If you are done configuring the firewall, enter `commit` from configuration mode.

Verify the Firewall Reboot

Confirm that the configuration is working properly.

Purpose

Verify the firewall reboot.

Action

From configuration mode, confirm firewall reboot by entering the `show system` command. If the output does not display the intended configuration, repeat the configuration instructions in this example. If you are done rebooting the firewall, enter `commit` from configuration mode.

Meaning

This `show system` command help verify whether the firewall rebooted.

Halt the SRX Series Firewalls

IN THIS SECTION

- [Requirements | 226](#)
- [Overview | 226](#)
- [Configuration | 226](#)
- [Verify the Firewall Halt | 228](#)

This example shows how to halt a device.

Requirements

Before halting the firewall, save and commit any Junos OS updates.

Overview

When the firewall is halted, all software processes stop and you can access the device through the console port only. Reboot the firewall by pressing any key on the keyboard.

If you cannot connect to the firewall through the console port, shut down the firewall by pressing and holding the power button on the front panel until the **POWER** LED turns off. After the firewall has shut down, you can power on the firewall by pressing the power button again. The **POWER** LED turns on during startup and remains steadily green when the firewall operates normally.

This example shows how to halt the system and stop software processes on the device immediately.

Configuration

IN THIS SECTION

- [Procedure | 227](#)

Procedure

CLI Quick Configuration

To configure this section quickly, follow these steps:

1. Copy the commands and paste them into a text file.
2. Remove any line breaks and adjust details to match your network.
3. Copy and paste the commands into the CLI at the [edit] level.
4. Enter commit from configuration mode.

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Review the "[Platform-Specific Halting Software Processes Behavior](#)" on [page 228](#) section for notes related to your platform.

From operational mode, enter:

```
user@host> request system halt at now
```

Step-By-Step Procedure

To halt an SRX Series Firewall immediately:

1. In the J-Web UI, select **Device Administration>Software Management>Reboot**.
2. Select **Halt Immediately**. After the firewall stops, you can access the firewall through the console port only.
3. Click **Schedule**. The J-Web UI displays a request to confirm the halt.
4. Click **OK** to confirm the operation. If the firewall halts, all software processes stop and you can access the firewall through the console port only. Reboot the firewall by pressing any key on the keyboard.
5. Click **OK** to check your configuration and save your changes as a candidate configuration.
6. If the system halt is successful, click **Commit Options>Commit**.

Results

Verify the Firewall Halt

Purpose

Verify the Firewall halt.

Action

From configuration mode, confirm system halt by entering the `show system` command. If the output does not display the intended configuration, repeat the configuration instructions in this example.

If the halt is successful, enter `commit` from configuration mode.

Meaning

The `show system` command helps verify whether the Firewall halts.

Platform-Specific Halting Software Processes Behavior

Use [Feature Explorer](#) to confirm platform and release support for specific features.

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

Platform	Difference
SRX Series	<ul style="list-style-type: none">SRX1500, SRX4100, and SRX4200 devices do not support the <code>request system halt</code> command used to halt the firewall and stop software processes on the firewall.

Manage Chassis Component Status on SRX Series Firewalls: Online and Offline Operations

SUMMARY

Learn how to manage and restart chassis component and verify the operational status on an SRX Series Firewall.

IN THIS SECTION

- [Restart the Chassis on SRX Series Firewalls | 230](#)

You can use the `request chassis` commands to bring chassis components (except Power Entry Modules and fans) online and offline.

To bring chassis components online and offline, enter these `request chassis` commands:

```
user@host> request chassis <fru> slot <slot#> pic <pic#> online
```

```
user@host> request chassis <fru> slot <slot#> pic <pic#> online
```

In each of the preceding `request chassis` commands, for SRX300, SRX320, SRX340, SRX345, and SRX380 Firewalls, `<fru>` can be `fpc`. With this command, you can change the Flexible PIC Concentrator (FPC) status.

- `fpc`—Changes the Flexible PIC Concentrator (FPC) status.

In each of the preceding `request chassis` commands, for SRX5800, SRX5600, and SRX5400 Firewalls, `<fru>` can be any of one of these values (?):

- `cb`—Changes the control board status.
- `fabric`—Changes the fabric status.
- `fpc`—Changes the Flexible PIC Concentrator (FPC) status.
- `fpm`—Changes the craft interface status.
- `pic`—Changes the physical interface card status.
- `routing-engine`—Changes the routing engine status.

The `request chassis` command is not supported for bringing SPCs online and offline.

Example:

To bring specific PIC and the corresponding FPC slot online, from operational mode, enter the following request chassis command:

```
user@host> request chassis pic pic-slot 1 fpc-slot 1 online
```

Restart the Chassis on SRX Series Firewalls

You can restart the chassis by using the `restart chassis-control` command with the following options:

- To restart the process gracefully:

```
user@host> restart chassis-control gracefully
```

- To restart the process immediately:

```
user@host> restart chassis-control immediately
```

- To restart the process softly:

```
user@host> restart chassis-control soft
```

Prepare the USB Flash Drive to Upgrade Junos OS on SRX300 Line of Firewall Devices

IN THIS CHAPTER

- [USB Flash Drives Usage on SRX Series Firewall Overview | 231](#)
- [Install Junos OS on SRX300 Line of Firewalls Using a USB Flash Drive | 233](#)

USB Flash Drives Usage on SRX Series Firewall Overview

SUMMARY

Learn how to format the drive, copy the Junos OS image, and enable automatic installation for remote upgrades and system recovery.

IN THIS SECTION

- [Prepare the USB Flash Drive | 232](#)

You can use the USB flash drive to simplify the upgrade of Junos OS images when you don't have console access to an SRX Series Firewall located at a remote site. This feature facilitates Junos OS image upgrade with minimum configuration. You need to only copy the image onto a USB flash drive, insert the drive into the USB port of the SRX Series Firewall, and perform a few simple steps. You can also use this feature to reformat a boot device and recover an SRX Series Firewall after boot media corruption.

All USB flash drives used on SRX Series Firewalls must have the following features:

- USB 2.0 or later
- Formatted with any one of these file systems:
 - -FAT
 - -FAT 32
 - -MA-DOS

Guidelines for USB Flash Drives

For the list of recommended USB storage device, see Knowledge Base article [KB31622](#).

The Junos OS package—for example, **junos-srxsme-15.1X49-D30.3-domestic.tgz**—on a USB device is commonly stored in the root drive as the only file.

Any USB memory product not listed as supported for SRX Series Firewalls has not been tested by Juniper Networks. The use of any unsupported USB memory product could cause unpredictable behavior in your SRX Series Firewall. Juniper Networks Technical Assistance Center (JTAC) can provide only limited support for issues related to unsupported hardware. We strongly recommend that you use only supported USB flash drives.

This feature is not supported on chassis clusters.

Prepare the USB Flash Drive

Before you begin:

- Copy the Junos OS upgrade image and its **autoinstall.conf** file to the USB storage device.
- Ensure that adequate space is available on the SRX Series Firewall to install the software image.

To prepare the USB flash drive and copy the Junos OS image onto the USB flash drive:

1. Insert the USB flash drive into the USB port of a PC or laptop running Windows.
2. From My Computer, right-click the drive Devices with Removable Storage.
3. Format the drive with the FAT or FAT32 file system.
4. Copy the Junos OS image onto the USB storage device.

For the installation process to succeed, copy only one image onto the USB storage device. In Junos OS Release 24.4R1, we renamed the software install images. Images named `junos-install-srxsme*` are recognized by the system to install Junos OS Release 24.4R1 or later releases. Images named `junos-srxsme*` are recognized by the system to install Junos OS Release 23.4R1 or previous releases.

5. Check the drive name detected in My Computer for the USB storage device. Open the command prompt window and type:

```
echo " " > <drive-name>:\autoinstall.conf
```

For example, if the drive detected is drive F, type `echo " " > F:\autoinstall.conf` at the command prompt. The empty file indicates to the system that the automatic installation of the Junos OS image from the USB storage device is supported.

6. (Optional) Create a text file named `junos-config.conf` and copy the file to the USB storage device. For example, the following file supports an automatic configuration update during the installation process:

```
system {
    host-name host-1;
    domain-name example.net;
    domain-search [ abc.exmaple.net example.net device1.example.net];
    root-authentication {
        encrypted-password "$ABC123"; ## SECRET-DATA
    }
}
...
...
routing-options {
    static {
        route 0.0.0.0/0 next-hop 10.207.31.254;
    }
}
```

You can use the file also as a backup configuration file for device recovery in case of upgrade failure or if the existing configuration is accidentally deleted.

Install Junos OS on SRX300 Line of Firewalls Using a USB Flash Drive

SUMMARY

Learn how to install Junos OS on SRX Series Firewalls using a USB flash drive. This process includes upgrade prerequisites, boot loader updates, and step-by-step installation procedures.

IN THIS SECTION

- [Install Junos OS on a Flash Drive | 234](#)
- [Platform-Specific USB Flash Drive Behavior | 236](#)

Prerequisites Before Installation

When you upgrade or downgrade SRX300 line of Firewalls to Junos OS Release 24.4R1 using a USB flash drive, the device reboots and comes up in amnesiac state. Therefore, before you install, ensure that you have saved the configuration file so that you can easily reconfigure the device using the console port if required.

Before you upgrade from either Junos OS Release 23.4R2-S3 or from Release 24.2R2 to Junos OS Release 24.4R1 for SRX300 Line of Firewalls, complete the following tasks:

- Upgrade the universal boot loader (U-Boot) software to at least version 3.15.
- Upgrade the loader to a build from the year 2023 or later. During the boot process, the loader reveals the build date. For example, this loader was built on May 23, 2023.

```
FreeBSD/mips U-Boot loader, Revision 2.0
(2023-05-23 22:48:57 builder@host)
```

Once you have installed either Junos OS Release 23.4R2-S3 or Release 24.2R2, the Junos OS image contains the latest boot loader binaries in the paths: **/boot/uboot** and **/boot/veloader**. You can upgrade the U-Boot software and veloader with the following steps:

1. From the CLI prompt, enter the `start shell` command.
2. From the shell prompt, update the U-Boot software with the `bootupgrade -u /boot/uboot` command.
3. From the shell prompt, update the veloader with the `bootupgrade -l /boot/veloader -x` command.
4. Reboot the device. When the device is up, you can use a USB flash drive to upgrade to Junos OS Release 24.4R1.

Install Junos OS on a Flash Drive

IN THIS SECTION

- [Additional Behavior During Installation | 235](#)

To install the Junos OS image on an SRX Series Firewall using a USB flash drive, follow the below steps:

1. Insert the USB flash drive into the USB port of the SRX Series Firewall and observe the LEDs.

The LEDs will initially blink amber and then steadily turn amber, indicating that the SRX Series device has detected the Junos OS image.

2. Press the **Reset Config** button on the SRX Series Firewall to initiate the installation process.

The LEDs glow steadily amber during installation.

3. Remove the USB flash drive.

Frequent plugging and unplugging of USB storage device is not supported. Always wait for device recognition before removing the drive.

4. Wait for the device to complete the installation.

- The SRX Series Firewall restarts automatically after installation.
- The LEDs glow green when the new Junos OS version is successfully installed.

If the LEDs do not change to amber:

- Press the **Power** button or restart the device.
- Wait for the LEDs to blink amber before attempting again.

During installation, if an error occurs, the LEDs turn red. Common reasons are:

Error Cause	Description
Corrupted Junos OS image	The image file on the USB flash drive is corrupted.
Configuration incompatibility	The existing configuration is incompatible with the new OS version.
Insufficient storage	Not enough space on the device to install the image.

In case of errors, you require console access to troubleshoot.

Additional Behavior During Installation

- When a USB storage device is inserted, the **Reset Config** button functions only for image upgrade.
- Other Reset Config button operations are overridden until the USB storage drive is removed.
- You do not need to wait for LEDs to turn steadily amber before pressing Reset Config; press the button when LEDs blink amber to avoid unnecessary delays.

You can use the `set system autoinstallation usb disable` command to prevent the automatic installation from the USB storage device. After using this command, if you insert the USB storage device into the USB port of the SRX Series Firewall, the installation process does not work.

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Review the ["Platform-Specific USB Flash Drive Behavior"](#) on page 236 section for notes related to your platform.

Platform-Specific USB Flash Drive Behavior

Use [Feature Explorer](#) to confirm platform and release support for specific features.

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

Platform	Difference
SRX Series	<ul style="list-style-type: none">SRX300 line of Firewalls that support USB flash drive do not support frequent plug and play of USB keys. You must wait for the device node creation before removing the USB key.

Upgrade and Install the Boot Loader on SRX Series Firewalls

IN THIS CHAPTER

- Upgrade the Bootloader on SRX Series Firewalls | 237
- Install Junos OS on SRX300 Line of Firewalls from the Boot Loader Using a TFTP Server | 239
- Install Junos OS on SRX300 Line of Firewalls from the Boot Loader Using a USB Storage Device | 242
- Upgrade the Software of SRX Series Firewalls by Using a PXE Boot Server | 243

Upgrade the Bootloader on SRX Series Firewalls

SUMMARY

Learn about installing Junos OS on SRX Series Firewalls using a USB flash drive, including prerequisites, bootloader updates, and step-by-step installation instructions.

To upgrade the bootloader to the latest version:

1. Upgrade to Junos OS Release 10.0 or later (with or without dual-root partitioning enabled).

The Junos OS 10.0 image contain the latest boot loader binaries in this path: `/boot/uboot`, `/boot/loader`.

2. Enter the shell prompt using the `start shell` command.
3. Run the following command from the shell prompt:

```
bootupgrade -u /boot/uboot -l /boot/loader
```

You can use the following commands to upgrade U-Boot or perform a cyclic redundancy check (CRC):

- `bootupgrade -s -u` — To upgrade the secondary boot loader.
- `bootupgrade -c u-boot` — To check CRC of the boot loader.

- `bootupgrade -s -c u-boot` — To check CRC for the secondary boot loader.
- `bootupgrade -c loader` — To check CRC for the loader on boot loader.

4. Enter the `show system firmware` command to check whether the upgrade is successful or not.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
FPC 1					
PIC 0	MLTE_FW	1	17.2.91	0	OK
Routing Engine 0	RE BIOS	0	3.8	3.6	OK
Routing Engine 0	RE BIOS Backup	1	3.6	3.6	OK

5. Reboot the system after upgrading the boot loader. Reboot is required for the new version to take effect.

When your device boots up, check the bootloader version in the console output:

```
scanning bus 0 for devices... 1 USB Device(s) found
  scanning usb for storage devices... 1 Storage Device(s) found

FreeBSD/MIPS U-Boot bootstrap loader, Revision 2.10
```

To verify the BIOS firmware version on the SRX Series Firewall, enter the `show chassis routing-engine bios` command.

```
user@host> show chassis routing-engine bios
Routing Engine BIOS Version: 1.5
```

Install Junos OS on SRX300 Line of Firewalls from the Boot Loader Using a TFTP Server

SUMMARY

Learn how to install Junos OS on SRX Series Firewalls using a TFTP server from the boot loader. This process includes prerequisites, U-Boot updates, and step-by-step installation procedures.

IN THIS SECTION

- [Before you begin | 239](#)

The SRX300 line of Firewalls are shipped with Junos OS loaded on the primary boot device. During Junos OS installation from the bootloader, the device retrieves the Junos OS package from a TFTP server. The internal media are then formatted, and the Junos OS image is installed.

You can use the bootloader to:

- Install Junos OS on the device for the first time.
- Recover the system from a file system corruption. You can install Junos OS from a TFTP server by using only the first onboard Ethernet interface. Installation from the loader-over-TFTP method does not work reliably over slow speeds or large latency networks.

Before you begin

Upgrade or downgrade the SRX300 Line of Firewall from Junos OS Release 24.4R1 using the boot loader. After rebooting, the device enters an amnesiac state. Therefore, before installing the software, you must save the configuration file so that you can easily reconfigure the device by using the console port.

Also, before upgrading from either Junos OS Release 23.4R2-S3 or 24.2R2 to Junos OS Release 24.4R1 for SRX300 Line of Firewalls, complete the following:

1. Upgrade the U-boot software to at least version 3.15. After you install either Junos OS Release 23.4R2-S3 or 24.2R2, the Junos OS image contains the latest bootloader binaries in these paths: **/boot/uboot** and **/boot/veloader**. You can upgrade the U-Boot software and veloader software as follows:
 - a. From the CLI prompt, enter the `start shell` command.
 - b. From the shell prompt, update the U-Boot software with the `bootupgrade -u /boot/uboot` command.
 - c. From the shell prompt, update the veloader with the `bootupgrade -l /boot/veloader -x` command.

- d. Reboot the device. Once the device is back up, you can use the Trivial File Transfer Protocol (TFTP) to upgrade to Junos OS Release 24.4R1.
- 2. Upgrade the bootloader to a build from the year 2023 or later. During the boot process, the bootloader reveals the build date. In the following example, you can see the build date as May 23, 2023.

```
FreeBSD/mips U-Boot loader, Revision 2.0
(2023-05-23 22:48:57 builder@host)
```

- 3. Verify these prerequisites:
 - You have access to the TFTP server with the Junos OS package to be installed.
 - The TFTP server supports BOOTP or DHCP. If the TFTP server does not support BOOTP or DHCP, you must set the environment variables before installing the software from the TFTP server.
 - Functional network connectivity exists between the device and the TFTP server over the first onboard Ethernet interface.

To install the Junos OS image on the internal media of the device:

- 1. Connect to the device by using the console connection and access the U-Boot prompt.
- 2. Reboot the device.

The following message and prompt appear:

```
Clearing DRAM..... done BIST check passed. Net:  pic init done (err = 0)octeth0 POST Passed
```

```
Press SPACE to abort autoboot in 3 seconds
```

- 3. Press the space bar to stop the automatic boot process.
The => U-Boot prompt appears.
- 4. Configure the environment variables listed in [Table 17 on page 240](#).

Table 17: Environment Variables Settings

Environment Variables	Description
gatewayip	IP address of the SRX Series Firewall

Table 17: Environment Variables Settings (Continued)

Environment Variables	Description
ipaddr	IP address of the SRX Series Firewall
netmask	Network mask
serverip	IP address of the TFTP server

You can see the configured environment variables in this example:

```

Clearing DRAM..... done
BIST check passed.
Net:  pic init done (err = 0)octeth0
POST Passed
Press SPACE to abort autoboot in 3 seconds
=>
=> setenv ipaddr 198.51.100.15
=> setenv netmask 255.255.255.0
=> setenv gatewayip 198.51.100.1
=> setenv serverip 203.0.113.2
=> saveenv

```

5. Reboot the system using the reset command.
6. Reboot the device.

The following message and prompt appear:

```
Loading /boot/defaults/loader.conf
```

Hit [Enter] to boot immediately, or space bar for command prompt.

7. Press the space bar to access the loader prompt (loader>).

At the loader> prompt, enter:

```
loader> install tftp://203.0.113.2/junos-srxsme-10.0R2-domestic.tgz
```

The URL has the format *tftp://tftp-serveripaddress/package* where the URL path is relative to the TFTP server's TFTP root directory.

When this command is executed:

1. The system downloads the Junos OS package from the TFTP server.
2. The system formats the internal media.
3. The system installs the Junos OS package on the formatted internal media.

After you install Junos OS, the device boots from the internal media. After the system boots up with Junos OS Release, you must upgrade the U-boot and boot loader immediately.

Install Junos OS on SRX300 Line of Firewalls from the Boot Loader Using a USB Storage Device

SUMMARY

Learn how to install Junos OS on SRX Series Firewalls from the boot loader using a USB storage device, including prerequisites, U-Boot updates, and step-by-step installation instructions.

When you use a USB flash drive on SRX300 Line of Firewalls for an upgrade to Junos OS Release 24.4R1 or a downgrade from this release, a firewall comes up in amnesiac state after reboot. Therefore, before installing the software, you must save the configuration file so that you can easily reconfigure the device by using the console port.

Before upgrading to Junos OS Release 24.4R1 or later for SRX300 Line of Firewalls, you must first upgrade the U-Boot software to at least version 3.15.

- Upgrade the bootloader to a build from the year 2023 or later. During the boot process, the bootloader reveals the build date. In the following example, you can see the build date as May 23, 2023.

```
FreeBSD/mips U-Boot loader, Revision 2.0  
(2023-05-23 22:48:57 builder@host)
```

To install Junos OS Release 24.4 or later from the bootloader using a USB storage device:

1. Format a USB storage device in MS-DOS format.
2. Copy the Junos OS image onto the USB storage device.
3. Plug the USB storage device into the SRX Series Firewall.

4. Stop the device at the loader prompt and issue the following command:

```
loader> install file:/// <image-path-on-usb>
```

An example of a command is as follows:

```
loader> install file:///junos-srxsme-10.0R2-domestic.tgz
```

This procedure formats the internal media and installs the new Junos OS image on the media with dual-root partitioning.

5. Remove the USB flash drive.

On SRX300 Line of Firewalls, frequent plug and play of USB keys is not supported. You must wait for the device node creation before removing the USB key.

If an installation error occurs, the LEDs turn red, indicating a corrupted Junos OS image on the USB flash drive. The error might also result from an incompatible SRX Series Firewall or insufficient space for the new image. You require console access to troubleshoot the error.

If the USB storage device is not recognized, you might see an error message such as Target device selected for installation: internal media cannot open package (error 2). In this case, power cycle the SRX Series Firewall with the USB device inserted and retry the installation using the bootloader.

Upgrade the Software of SRX Series Firewalls by Using a PXE Boot Server

SUMMARY

Learn how to upgrade the software of SRX Series Firewalls (SRX1500, SRX4100, and SRX4600) using a PXE boot server. This topic covers the step-by-step process, including preparing the PXE server, copying the installation image, and rebooting the device to complete the upgrade.

IN THIS SECTION

- [Upgrade SRX1500 Firewall Software | 243](#)
- [Upgrade SRX4200 Firewall Software | 247](#)
- [Upgrade the Software of SRX4600 Device | 250](#)

Upgrade SRX1500 Firewall Software

The build image that you install on the device defines the device software version. You can change the version of the device by upgrading it.

You can upgrade the software of a device by using the Preboot Execution Environment (PXE) boot server. A PXE boot prepares a client/server environment to boot devices by using a network interface that is independent of available data storage devices or installed operating systems. The TFTP server stores the operating system image. You can have a separate PXE boot server for each image.

To copy the required image to the PXE boot server and install the image on your firewall:

1. Remove the previously installed files, if any, from the **/var/lib/tftpboot/** directory.

```
user@host> rm -f /tftpboot
user@host> mkdir /tftpboot
```

2. Copy the downloaded installation media to the **/var/lib/tftpboot/** directory in the PXE boot server. For example:

```
scp /volume/build/junos/20.1/release/zyx/ship/
junos-install-media-pxe-srxentedge-x86-64-20.3I-20200517_dev_common.0.1613.tgz
user@host:/var/lib/tftpboot/
```

3. Log in to the PXE boot server and verify the installation file.

For example:

```
user@host> ls -lh junos-install-media-pxe-srxentedge-
x86-64-20.3I-20200517_dev_common.0.1613.tgz
-rw-r--r-- 1 root root 1.8G June 08 00:42 junos-install-media-pxe-srxentedge-
x86-64-20.3I-20200517_dev_common.0.1613.tgz
```

4. Extract the **junos-install-media-pxe-srxentedge** TAR file.

For example:

```
user@host> tar xvfz junos-install-media-pxe-srxentedge-
x86-64-20.3I-20200517_dev_common.0.1613.tgz -C /var/lib

./initramfs.cpio.gz
./initrd.cpio.gz
./upgrade_platform
./initramfs.cpio.gz.psig
./vmlinuz.psig
./HOST_COMPAT_VERSION
./application-pkg.tgz
./EFI/
```

```

./EFI/BOOT/
./EFI/BOOT/BOOTX64.EFI
./EFI/BOOT/grub-root.pub
./EFI/BOOT/grub-trusted.gpg.psig
./EFI/BOOT/grub-trusted.gpg
./linux.checksum
./version.txt
./host-version
./vmlinuz

```

5. Copy the BOOTX64.EFI file to the TFTP home folder (`/var/lib/tftpboot/`).

```

user@host> cp EFI/BOOT/BOOTX64.EFI /var/lib/tftpboot/

```

6. Create a secure boot folder at `/var/lib/tftpboot/`.

```

user@host> rm -rf /var/lib/tftpboot/secure-boot
user@host> mkdir /var/lib/tftpboot/secure-boot

```

7. Copy the grub files in the **secure-boot** folder.

```

user@host> cp EFI/BOOT/grub-root.pub secure-boot/
user@host> cp EFI/BOOT/grub-trusted.gpg secure-boot/
user@host> cp EFI/BOOT/grub-trusted.gpg.psig secure-boot/

```

8. Move **initrd.cpio.gz** and **application-pkg.tgz** in FTP server folder (`/var/ftp/`).

```

user@host> mv application-pkg.tgz /var/ftp/
user@host> mv initrd.cpio.gz /var/ftp/

```

9. Create **grub-startup.cfg** in the `/var/lib/tftpboot/secure-boot` folder.

```

user@host> cat grub-startup.cfg
insmod search
insmod linux
insmod tftp
insmod reboot
insmod efi_gop
insmod efi_uga
insmod read

```

```

insmod chain
insmod boot
insmod font
insmod serial

set timeout=5

menuentry 'PXE image' {
    set net_default_server=192.168.120.1
    echo 'Loading ...'
    linux (tftp)/vmlinuz root=/dev/ram quiet console=ttyS0,9600n8 acpi=ht
    libata.force=noncq acpi_enforce_resources=lax install rootfs=ftp://192.168.120.1/
    initrd.cpio.gz app_pkg=ftp://192.168.120.1/application-pkg.tgz efi=debug intel_iommu=on
    isolcpus=2,3
    echo 'Loading initial ramdisk ...'
    initrd (tftp)/initramfs.cpio.gz
}

```

10. After you copy the image to the PXE boot server, reboot the device to install the image.

```

user@host> request system reboot

```

The firewall boots from the PXE server and installs the image on both the Solid State Drive (SSDs).

If the device fails to reboot, you can install the software by using the USB flash disk. However, if the firewall fails to reboot or is inaccessible after you use this method, perform these steps using the console connection:

1. Reboot or power on the device
2. Press the **ESC** button to go to the Boot Manager menu.
3. Select Setup Utility and then press Enter.
4. Select the boot options:
 - Boot type—UEFI Boot Type,
 - PXE boot capability—UEFI:IPv4,
 - First boot device—PXE on ME,
 - Enable the network stack option.
5. Press F10.

6. Verify that the upgrade is successful by entering the `show version` command from the operational mode. If you have upgraded the software of the device to an SRX1500, the new version of the device is srx1500.

```
user@host> show version
Hostname: host
Model: srx1500
```

Juniper Networks does not support using the `request system software rollback` command to revert to the previously installed software.

Upgrade SRX4200 Firewall Software

The build image that you install on the device defines the device's software version. You can change the software version of the device by upgrading it.

You can upgrade the device software by using the Preboot Execution Environment (PXE) boot server. A PXE boot prepares a client/server environment to boot devices by using a network interface that is independent of available data storage devices or installed operating systems. The Trivial File Transfer Protocol (TFTP) server stores the operating system image. You can have a separate PXE boot server for each image.

To copy the required image to the PXE boot server and install the image on your firewall:

1. Remove the previously installed files, if any, from the **`var/lib/tftpboot/`** directory.

```
user@host> rm -f /tftpboot
user@host> mkdir /tftpboot
```

2. Copy the downloaded installation media to the **`/var/lib/tftpboot/`** directory in the PXE boot server. For example:

```
scp /volume/build/junos/20.1/release/zyx/ship/
junos-install-media-pxe-srxmr-x86-64-20.3I-20200520_dev_common.0.1928.tgz
user@host:/var/lib/tftpboot/
```

3. Log in to the PXE boot server and verify the installation file.

For example:

```
user@host> ls -lh junos-install-media-pxe-srxmr-x86-64-20.3I-20200520_dev_common.0.1928.tgz
-rw-r--r-- 1 root root 1.8G Jun 08 00:42 junos-install-media-pxe-srxmr-
x86-64-20.3I-20200520_dev_common.0.1928.tgz
```

4. Extract the **junos-install-media-pxe-srxmr** TAR file.

For example:

```
user@host> tar xvzf junos-install-media-pxe-srxmr-
x86-64-20.3I-20200520_dev_common.0.1928.tgz -C /var/lib

./initramfs.cpio.gz
./initrd.cpio.gz
./upgrade_platform
./initramfs.cpio.gz.psig
./vmlinuz.psig
./HOST_COMPAT_VERSION
./application-pkg.tgz
./EFI/
./EFI/BOOT/
./EFI/BOOT/BOOTX64.EFI
./EFI/BOOT/grub-root.pub
./EFI/BOOT/grub-trusted.gpg.psig
./EFI/BOOT/grub-trusted.gpg
./linux.checksum
./version.txt
./host-version
./vmlinuz
```

5. Move **initrd.cpio.gz** and **application-pkg.tgz** in TFTP server folder (**/var/ftp/**).

```
user@host> mv application-pkg.tgz /var/ftp/
user@host> mv initrd.cpio.gz /var/ftp/
```

6. Install **syslinux** on the TFTP server.

```
user@host> yum install syslinux
```


7. Copy syslinux files to the TFTP server.

```
user@host> cp /usr/share/syslinux/menu.c32 /usr/share/syslinux/vesamenu.c32 /usr/share/
syslinux/pxelinux.0 /var/lib/tftpboot/
```

8. Create PXE menu.

```
user@host> mkdir /var/lib/tftpboot/pxelinux.cfg
```

9. Create a new default file at PXE menu.

```
user@host> cat pxelinux.cfg/default
default vesamenu.c32
prompt 0
timeout 800

#display boot.msg

#menu background splash.jpg
menu title Welcome!
menu color border 0 #ffffff #00000000
menu color sel 7 #ffffff #ff000000
menu color title 0 #ffffff #00000000
menu color tabmsg 0 #ffffff #00000000
menu color unsel 0 #ffffff #00000000
menu color hotssel 0 #ff000000 #ffffff
menu color hotkey 7 #ffffff #ff000000
menu color scrollbar 0 #ffffff #00000000

LABEL SRXMR---20.3
    MENU LABEL ^B SRXMR---20.3
    KERNEL vmlinuz
    INITRD initramfs.cpio.gz
    APPEND vm console=ttyS0,9600n8 root=/dev/ram intel_iommu=on acpi=off isolcpus=2,3
libata.force=noncq acpi_enforce_resources=lax install rootfs=ftp://192.168.120.1/
initrd.cpio.gz install app_pkg=ftp://192.168.120.1/application-pkg.tgz
```

10. Copy the image to the PXE boot server and reboot the device to install the image.

```
user@host> request system reboot
```

The firewall boots from the PXE bootserver and installs the image on both the SSDs.

If the device fails to reboot, you can install the software by using the USB disk. However, if the firewall fails to reboot or is inaccessible after you use this method, perform these steps using the console connection:

1. Reboot or power on the device
2. Press the **ESC** button to go to the Boot Manager menu.
3. Select the boot options:
 - Boot mode—LEGACY,
 - Boot option 1—NETWORK,
 - Disable the network stack option.
4. Click save and exit or press F4 to start PXE boot.
5. Select the menu from the screen and click **Enter** to reboot the device.
6. Select Hard Disk as boot option 1.
7. Click save and exit or press F4 to start PXE boot.
8. From operational mode, verify that the upgrade is successful. If you have upgraded the version of the device to an SRX4100, the new version of the device is `srx4100`.

```
user@host> show version
Hostname: host
Model: srx4100
```

Juniper Networks does not support using the `request system software rollback` command to revert to the previously installed software version.

Upgrade the Software of SRX4600 Device

The build image loaded on the device defines the software of the device. You can change the software of the device by upgrading it.

You can upgrade the software version of a device by using the Preboot Execution Environment (PXE) boot server. A PXE boot prepares a client/server environment to boot devices by using a network interface that is independent of available data storage devices or installed operating systems. The image of the operating system is stored on a Trivial File Transfer Protocol (TFTP) server. You can have a separate PXE boot server for each image.

To copy the image to the PXE boot server and install the image:

1. Remove the previously installed files, if any, from the `/var/lib/tftpboot/` directory.

```
user@host> rm -f /tftpboot
user@host> mkdir /tftpboot
```

2. Copy the downloaded installation media to the `/var/lib/tftpboot/` directory in the PXE boot server. For example:

```
scp /volume/build/junos/20.1/release/zyx/ship/
junos-install-media-pxe-srxhe-x86-64-20.3I-20200521_dev_common.0.1013.tgz
user@host:/var/lib/tftpboot/
```

3. Log in to the PXE boot server and verify the installation file. For example:

```
user@host> ls -lh junos-install-media-pxe-srxhe-x86-64-20.3I-20200521_dev_common.0.1013.tgz
-rw-r--r-- 1 root root 1.8G June 08 00:42 junos-install-media-pxe-srxhe-
x86-64-20.3I-20200521_dev_common.0.1013.tgz
```

4. Extract the `junos-install-media-pxe-srxhe` TAR file. For example:

```
user@host> tar xvzf junos-install-media-pxe-srxhe-
x86-64-20.3I-20200521_dev_common.0.1013.tgz -C /var/lib

./initramfs.cpio.gz
./initrd.cpio.gz
./upgrade_platform
./initramfs.cpio.gz.psig
./vmlinuz.psig
./HOST_COMPAT_VERSION
./application-pkg.tgz
./EFI/
```

```

./EFI/BOOT/
./EFI/BOOT/BOOTX64.EFI
./EFI/BOOT/grub-root.pub
./EFI/BOOT/grub-trusted.gpg.psig
./EFI/BOOT/grub-trusted.gpg
./linux.checksum
./version.txt
./host-version
./vmlinuz

```

5. Copy the **BOOTX64.EFI** file to the TFTP home folder (**/var/lib/tftpboot/**).

```

user@host> cp EFI/BOOT/BOOTX64.EFI /var/lib/tftpboot/

```

6. Create a secure boot folder at **/var/lib/tftpboot/**.

```

user@host> rm -rf /var/lib/tftpboot/secure-boot
user@host> mkdir /var/lib/tftpboot/secure-boot

```

7. Copy the grub files in the **secure-boot** folder.

```

user@host> cp EFI/BOOT/grub-root.pub secure-boot/
user@host> cp EFI/BOOT/grub-trusted.gpg secure-boot/
user@host> cp EFI/BOOT/grub-trusted.gpg.psig secure-boot/

```

8. Move **initrd.cpio.gz** and **application-pkg.tgz** in the FTP server folder (**/var/ftp/**)

```

user@host> mv application-pkg.tgz /var/ftp/
user@host> mv initrd.cpio.gz /var/ftp/

```

9. Create **grub-startup.cfg** in the **/var/lib/tftpboot/secure-boot** folder.

```

user@host> cat grub-startup.cfg
insmod search
insmod linux
insmod tftp
insmod reboot
insmod efi_gop
insmod efi_uga
insmod read

```

```

insmod chain
insmod boot
insmod font
insmod serial

set timeout=5

menuentry 'PXE image' {
    set net_default_server=192.168.120.1
    echo 'Loading ...'
    linux (tftp)/vmlinuz root=/dev/ram quiet console=ttyS0,9600n8 acpi=ht
    libata.force=noncq acpi_enforce_resources=lax install rootfs=ftp://192.168.120.1/
    initrd.cpio.gz app_pkg=ftp://192.168.120.1/application-pkg.tgz efi=debug intel_iommu=on
    isolcpus=2,3
    echo 'Loading initial ramdisk ...'
    initrd (tftp)/initramfs.cpio.gz
}

```

10. After you copy the image to the PXE boot server, reboot the device to install the image.

```

user@host> request system reboot

```

The firewall boots from the PXE server and installs the image on both the SSDs.

If the device fails to reboot, you can install the software by using the USB disk. However, if the firewall fails to reboot or is inaccessible after you use this method, perform these steps using the console connection:

1. Reboot or power on the device.
2. Press the **ESC** button to go to the Boot Manager menu.
3. Select Setup Utility and then press Enter.
4. Select the PXE boot capability as UEFI:IPv4, disable HDD, and enable ETH00 under **EPI**.
5. Click F10.

6. From operational mode, verify that the upgrade is successful. If you have upgraded the software version of the device to an SRX4600, the new version of the device is srx4600.

```
user@host> show version
Hostname: host
Model: srx4600
```

Juniper Networks does not support using the `request system software rollback` command to revert to the previously installed software version.

Configure Root Partitions on SRX Series Devices

SUMMARY

Learn how to configure root partitions on SRX Series Firewalls, including the benefits of dual-root partitioning for improved reliability and recovery. This topic covers the boot process, automatic recovery with the auto-snapshot feature, and steps to reinstall single-root partitioning for older Junos OS versions.

IN THIS SECTION

- [Dual-Root Partition on SRX Series Firewalls | 254](#)
- [Reinstall Single-Root Partition on SRX Series Firewalls | 260](#)
- [Platform-Specific Storage Media Behavior | 260](#)

Dual-Root Partition on SRX Series Firewalls

IN THIS SECTION

- [Boot Media and Boot Partition on SRX Series Firewalls | 255](#)
- [Key Functionality of the Dual-Root Partition | 255](#)
- [Automatic Recovery of the Primary Junos OS Image with Dual-Root Partitioning | 256](#)
- [How the Primary Junos OS Image with Dual-Root Partitioning Recovers Devices | 257](#)
- [How Junos OS Release 10.0 or Later Upgrades with Dual-Root Partitioning | 259](#)

Dual-root partitioning ensures that an SRX Series Firewall remains functional even when the file system gets corrupted and facilitates easy recovery of the file system.

In single-root partitioning as both the primary and backup Junos OS images are in the same root partition, the system fails to boot if the root file system gets corrupted. Dual-root partitioning prevents this situation by keeping the primary and backup Junos OS images in two independently bootable root partitions. If the primary root partition becomes corrupted, the system can still boot from the backup Junos OS image located in the other root partition and remain fully functional.

Boot Media and Boot Partition on SRX Series Firewalls

When the SRX Series Firewall powers on, it tries to boot the Junos OS from the default storage media. If the device fails to boot from the default storage media, the device tries to boot from the alternate storage media.

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Review the ["Platform-Specific Storage Media Behavior" on page 260](#) section for notes related to your platform.

With dual-root partitioning, the SRX Series Firewall first tries to boot Junos OS from the primary root partition and then from the backup root partition on the default storage media.

If Junos OS fails to boot from both the primary and backup root partitions of a media, the firewall tries to boot the software from the next available storage media. The firewall remains fully functional even if it boots Junos OS from the backup root partition of the storage media.

Key Functionality of the Dual-Root Partition

Dual-root partitioning has the following important features:

- The primary and backup copies of Junos OS images reside in separate partitions. The partition containing the backup copy is mounted only when required. With the single-root partitioning scheme, there is one root partition in the default that contains both the primary and backup Junos OS images.
- The `request system software add` command for a Junos OS package erases the contents of the other root partition. The contents of the other root partition will not be valid unless software installation is completed successfully.

- You can reinstall add-on packages, such as jais or jfirmware, as required after a new Junos OS image is installed.
- The request system software rollback command does not delete the current Junos OS image. You revert to the image by issuing the rollback command.
- The request system software delete-backup and request system software validate commands do not result in any action.

Automatic Recovery of the Primary Junos OS Image with Dual-Root Partitioning

The automatic snapshot feature repairs the corrupted primary root partition when the device reboots from the alternate root. This repair is accomplished by taking a snapshot of the alternate root onto the primary root automatically rather than manually from the CLI.

When this feature is enabled, the device performs the following actions to reboot from the alternate root (because of a corrupted primary root or power cycle during restart):

1. Displays a prominent message indicating a failure to boot from the primary root.

```
*****
**                                                                 **
**  WARNING: THIS DEVICE HAS BOOTED FROM THE BACKUP JUNOS IMAGE  **
**                                                                 **
**  It is possible that the primary copy of JUNOS failed to boot up **
**  properly, and so this device has booted from the backup copy.  **
**                                                                 **
**  Please re-install JUNOS to recover the primary copy in case   **
**  it has been corrupted and if auto-snapshot feature is not     **
**  enabled.                                                       **
**                                                                 **
*****
```

2. A system boot from backup root alarm is set. This action is useful for devices that do not have console access.
3. A snapshot of the alternate root onto the primary root is made.
4. Once the snapshot is complete, the system boot from backup root alarm is cleared.

During the next reboot, the system determines the good image on the primary root and boots normally. Perform the snapshot once all the processes start. This is done to avoid any increase in the reboot time.

By default the automatic snapshot feature is disabled. If you do not maintain the same Junos OS release in both the partitions, ensure that the automatic snapshot feature remains disabled. For instance, the alternate partition might have an earlier Junos OS release. If the device reboots from the alternate root partition, the automatic snapshot feature replaces the later Junos OS release with the earlier release. When automatic snapshot is disabled and the system reboots from the alternate root partition, you receive an alarm indicating that the system has rebooted from its alternate partition.

Given you maintain the same Junos OS release on both the root partitions, enable this feature with the `set system auto-snapshot` command. After this feature ensures recovery of the primary root partition, the device successfully boots from the primary root partition on the next reboot.

Execute the `delete system auto-snapshot` command to delete all backed-up data and disable automatic snapshot, if required.

Use the `show system auto-snapshot` command to check the status of automatic snapshot.

When automatic snapshot is in progress, you cannot run a manual snapshot command concurrently. If you attempt this action, the following error message appears:

```
Snapshot already in progress. Please try after sometime.
```

If you log into the device when the automatic snapshot feature is in progress, the following banner appears: The device has booted from the alternate partition, auto-snapshot is in progress.

How the Primary Junos OS Image with Dual-Root Partitioning Recovers Devices

If the SRX Series Firewall fails to boot from the primary Junos OS image and boots up from the backup image, you see the following message on the console interface. The message appears at the time of login, informing you about device bootup using the alternate image.

```
login: user
```

```
Password:
```

```
*****
```

```
**
```

```
**
```

```
** WARNING: THIS DEVICE HAS BOOTED FROM THE BACKUP JUNOS IMAGE
```

```
**
```

```
**
```

```
**
```

```

** It is possible that the active copy of JUNOS failed to boot up **
** properly, and so this device has booted from the backup copy. **
**                                                                **
** Please re-install JUNOS to recover the active copy in case **
** it has been corrupted. **
**                                                                **
*****

```

Because the system is left with only one functional root partition, you must immediately restore the primary Junos OS image using one of the following methods:

- Install a new image using the CLI or J-Web user interface. The newly installed image becomes the primary image, and the device boots from this image on the next reboot.
- Use a snapshot of the backup root partition by entering the `request system snapshot slice alternate` command. After the primary root partition is recovered using this method, the device successfully boots from the primary root partition on the next reboot. After the procedure, the primary root partition contains the same version of Junos OS as the backup root partition. After the automatic snapshot process is complete, the system boot from backup root alarm is cleared.

You can use the command `request system snapshot slice alternate` to back up the currently running root file system (primary or secondary) to the other root partition. With this command, you also:

- Save an image of the primary root partition in the backup root partition when the system boots from the primary root partition.
- Save an image of the backup root partition in the primary root partition when the system boots from the backup root partition.

The process of restoring the alternate root by using the CLI command `request system snapshot slice alternate` takes several minutes to complete. If you terminate the operation before completion, the alternate root might not have all the required contents to function properly.

How Junos OS Release 10.0 or Later Upgrades with Dual-Root Partitioning

To format the media with dual-root partitioning while upgrading to Junos OS Release 10.0 or later, use one of the following installation methods:

- Installation from the boot loader using a Trivial File Transfer Protocol (TFTP) server. We recommend using this method if console access to the system is available and a TFTP server is available in the network.
- Installation from the boot loader using a USB storage device. We recommend using this method if console access to the system is available and the system can be physically accessed to plug in a USB storage device.
- Installation from the CLI using the `partition` option. We recommend using this method only if console access is not available. You can perform this installation remotely.

Dual-Root and Single-Root Partition

Some Junos OS upgrade methods format the internal media before Junos OS installation. You must use one of these upgrade methods to install Junos OS with dual-root partitioning.

These upgrade methods format the internal media before Junos OS installation:

- Installation from the boot loader using a TFTP server
- Installation from the boot loader using a USB storage device
- Installation from the CLI using the `partition` option (available in Junos OS Release 10.0 and later)
- Installation using the J-Web user interface

These upgrade methods retain the existing partitioning scheme:

- Installation using the CLI without the `partition` option
- Installation using the J-Web user interface

Upgrade methods that format the internal media before installation wipe out the existing contents of the media. Only the current configuration is preserved. You must back up any important data.

After the media is formatted with the dual-root partitioning scheme, you can use conventional CLI or J-Web user interface installation methods for subsequent upgrades. These methods retain the existing partitioning and contents of the media.

Reinstall Single-Root Partition on SRX Series Firewalls

To reinstall the single-root partition on SRX Series Firewalls, you need to consider the compatibility between Junos OS releases and partitioning schemes. Junos OS Release 9.6 and earlier supports only single-root partitioning, whereas later releases support dual-root partitioning. If you attempt to install Junos OS Release 9.6 or earlier on a device with dual-root partitioning without reformatting the internal media, the installation will fail.

To reinstall the single-root partition:

1. Reformat the media using the command:

```
user@host>request system software add partition
```

2. Reboot the device after the Junos OS installation by using the command:

```
user@host>request system reboot
```

The previous software release gets installed after you reboot the device. Using the partition option erases the dual-root partitioning scheme, removing access to dual-root partitioning features such as improved rollback and recovery.

Platform-Specific Storage Media Behavior

Use [Feature Explorer](#) to confirm platform and release support for specific features.

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

Platform	Difference
SRX Series	<ul style="list-style-type: none"> • SRX300, SRX320, SRX340 and SRX345 devices support eUSB disk (default) and USB storage device (alternate). • SRX380 device supports internal SSD (default) and USB storage device (alternate). • SRX300, SRX320, SRX340, SRX345, and SRX380 devices support the automatic snapshot feature.

RELATED DOCUMENTATION

| [Install Software on SRX Series Devices](#)

CHAPTER 12

Veriexec and OpenPGP

IN THIS CHAPTER

- [Veriexec File Signing and Verification for Junos OS | 262](#)
- [Generate OpenPGP Keys for Secure Package Signing on Junos OS | 268](#)

Veriexec File Signing and Verification for Junos OS

SUMMARY

Learn about Veriexec file-signing and verification, benefits and how to use it on SRX Series Firewall.

IN THIS SECTION

- [How Veriexec Works | 263](#)
- [Benefits of Veriexec | 264](#)
- [How to Verify If Veriexec Works – Option 1 | 264](#)
- [Install Veriexec Loader | 265](#)

Verified Exec (also known as veriexec) is a file-signing and verification scheme that protects the Junos operating system (OS) against unauthorized software and activity that might compromise the integrity of your device. Originally developed for the NetBSD OS, veriexec was adapted for Junos OS and is enabled by default.

Authorized files, that is certain files that ship with Junos OS, has an associated fingerprint that veriexec checks to determine whether the file can be used (executed or even opened). Any file which lacks a valid fingerprint cannot be executed or read by applications that require verified input.

The `/bin/sh` file does not require verified input. You can use this file to run arbitrary scripts. From a risk perspective, these scripts are the same as interactive commands, which are controlled through user authentication and permissions. However, if a verified shell script contains instructions to run an arbitrary script, that is, a file without a signature in the manifest, execution of that file is prevented.

How Veriexec Works

Veriexec provides the kernel with a digitally signed manifest consisting of a set of fingerprints for all the executables and other files that must remain immutable. The veriexec loader feeds the contents of the manifest to the kernel only if associated digital signatures of the manifest are successfully verified. The kernel can then verify if a file matches its fingerprint. If veriexec is enforced, only executables with a verified fingerprint will run. The protected files cannot be written to, modified, or changed.

Each installation image contains a manifest, which is a read-only file. It contains entries such as the following:

```
etc/rc sha1=478eeda6750c455fbfc18eeb06093e32a341911b uid=0 gid=0 mode=644
etc/rc.verify sha1=15566bb2731abee890fabd0ae8799e02071e006c uid=0 gid=0 mode=644

usr/libexec/veriexec-ext.so.1 sha1=8929292d008d12cd5beb2b9d9537458d4974dd22 uid=0 gid=0 mode=550
no_fips

sbin/verify-sig sha1=cd3ffd45f30f1f9441e1d4a366955d8e2c284834 uid=0 gid=0 mode=555 no_ptrace
sbin/veriexec sha1=7b40c1eae9658f4a450eb1aa3df74506be701baf uid=0 gid=0 mode=555 no_ptrace

jail/usr/bin/php sha1=c444144fef5d65f7bbc376dc3ebb24373f1433a2 uid=0 gid=0 mode=555 indirect
no_fips

usr/sbin/chassisd sha1=61b82b36da9c6fb7eeb413d809ae2764a8a3cebc uid=0 gid=0 mode=555 trusted
```

The log message is in the following format:

```
/kernel:veriexec:fingerprintfordev<deviceid>,file<fileid><calculatedfingerprint>!=
<fingerprintinthemanifest>
```

If a file has been modified and the resulting fingerprint differs from the one in the manifest, you see a log message, such as the following example:

```
/kernel:veriexec:fingerprintfordev100728577,file70750 64ea873ed0ca43b113f87fa25fb30f9f60030cec!=
0d9457c041bb3646eb4b9708ba605facb84a2cd0
```

Fingerprint mismatch indicates that the file has been modified. Don't try to run such a file, as it could contain corrupted code. Contact JTAC for guidance.

Benefits of Veriexec

- Secure systems—Safeguard Juniper Networks routers, switches, and firewalls from security breaches.
- Prevent unauthorized access—Block threat actors from gaining persistent, unauthorized access or causing system failure.
- Prevent malware execution—Block unauthorized modifications and malware through prevention of unsigned binary execution.
- Support authorized code—Add signed, authorized code to Junos OS with veriexec enforcement using the JET SDK. For more information about the SDK solution, see [Juniper Extension Toolkit Developer Guide](#).

How to Verify If Veriexec Works — Option 1

Some Junos OS platforms offer an optional version of Junos OS with veriexec enforcement disabled. For detailed information, see [Junos Enhanced Automation](#).

Administrators can check whether veriexec is enforced by running the following commands from the Junos OS CLI shell:

1. Start the shell.

```
username@hostname> start shell
%
```

2. Use the `sysctl security.mac.veriexec.state` command for Junos OS Release 15.1 and Later

```
% sysctl security.mac.veriexec.state
security.mac.veriexec.state: loaded active enforce
%
```

If veriexec is enforced, the output is `security.mac.veriexec.state: loaded active enforce`. If veriexec is not enforced, the output is `security.mac.veriexec.state: loaded active`.

How to Verify If Veriexec Works — Option 2

You can confirm whether veriexec is working by copying an authorized file (for example, `/usr/bin/id`) to a new location as shown below. Veriexec prevents the operation, because even though the file is identical, a valid fingerprint exists only for `/usr/bin/id`, not for `/tmp/id`. To verify the file integrity, veriexec

evaluates the underlying Linux properties of the file. These properties, rather than the file itself, change after you copy the file to another location.

1. Start the shell.

```
username@hostname> start shell
#
```

2. Change directories and then copy an authorized file, for example, `/usr/bin/id`, to a new location.

```
# /usr/bin/id
uid=928(username) gid=20 groups=20,0(wheel),10(field)
# cp /usr/bin/id /tmp
```

Results

If veriexec is being enforced, an authentication error message is generated. If an error message is not generated, the file will run as normal.

Output generated when veriexec is enforced, showing the file is blocked:

```
# /tmp/id
/bin/sh: /tmp/id: Authentication error
#
```

Output generated when veriexec is not enforced, showing the file is copied

```
# /tmp/id
#
```

Install Veriexec Loader

The veriexec loader validates the Junos OS image that you install from a Trivial File Transfer Protocol (TFTP) server or a USB storage device.

- To install the Junos OS image from a TFTP server use this command:

```
loader> install tftp://[host]/package
```

- Install the Junos OS image from a USB storage device using the following command.

```
loader> install file:///package
```

The veriexec-capable loader validates the Junos OS image. The loader boots up only new Junos OS images with fingerprints and does not boot existing images without fingerprints or kernel. You can use the `nextboot` function to check the current bootup device.

```
username@hostname# nextboot
Platform: srx-sword
  eUSB
usb
current bootdev is: eUSB
```

Bootupgrade is a tool available in the Junos OS package to support BIOS firmware upgrading. You can use the `bootupgrade` command to upgrade, check U-Boot, manually load, and install the veriexec-capable loader. The `bootupgrade -c loader` command prints the version string for the current loader.

Before you install the veriexec-capable loader to a Junos OS image, identification of Junos OS fingerprints is carried out in both dual-root partitions. The veriexec capable loader can be installed only when both dual-root partitions have Junos OS with fingerprints.

Install the veriexec-capable loader from the Junos OS CLI shell:

1. Start the shell.

```
username@hostname> start shell
%
```

2. Use the `bootupgrade -l /boot/veloder` command to install the veriexec-capable loader.

```
% bootupgrade -l /boot/veloder
Checking Loader CRC... veloder size 1251641 OK
```

3. You can see different scenarios here:

- Use request system software add /var/tmp/xxx.tgz no-copy no-validate command to install Junos OS with fingerprints.

```
username@hostname> request system software add /var/tmp/junos-
srxsme-20.4I-20200810_dev_common.0.0833.tgz no-copy no-validate
Formatting alternate root (/dev/ad0s2a)...
/dev/ad0s2a: 600.0MB (1228732 sectors) block size 16384, fragment size 2048
    using 4 cylinder groups of 150.00MB, 9600 blks, 19200 inodes.
super-block backups (for fsck -b #) at:
32, 307232, 614432, 921632
Installing package '/altroot/cf/packages/install-tmp/
junos-20.4I-20200810_dev_common.0.0833' ...
Verified junos-boot-srxsme.tgz signed by PackageDevelopmentECP256_2020 method
ECDSA256+SHA256
Verified junos-srxsme-domestic signed by PackageDevelopmentECP256_2020 method
ECDSA256+SHA256
Verified manifest signed by PackageDevelopmentECP256_2020 method ECDSA256+SHA256

WARNING:    The software that is being installed has limited support.
WARNING:    Run 'file show /etc/notices/unsupported.txt' for details.

JUNOS 20.4I-20200810_dev_common.0.0833 will become active at next reboot
WARNING: A reboot is required to load this software correctly
WARNING:    Use the 'request system reboot' command
WARNING:    when software installation is complete
Saving state for rollback ...
```

- If the verixec-capable loader doesn't support the target Junos OS images of previous releases, you can downgrade to an earlier version of the the loader compatible with the release. Use the request system software add /var/tmp/xxx.tgz no-copy no-validate command to automatically downgrade to an earlier version of the loader.

```
username@hostname> request system software add /var/tmp/junos-srxsme-19.4R1.3.tgz no-copy
no-validate
WARNING: Package junos-19.4R1.3 version 19.4R1.3 is not compatible with current loader
WARNING: Automatic recovering loader, please wait ...
Upgrading Loader...
#####
Verifying the loader image... OK
WARNING: The new boot firmware will take effect when the system is rebooted.
WARNING: Loader recover finish.
```

```

Formatting alternate root (/dev/ad0s1a)...
/dev/ad0s1a: 598.5MB (1225692 sectors) block size 16384, fragment size 2048
    using 4 cylinder groups of 149.62MB, 9576 blks, 19200 inodes.
super-block backups (for fsck -b #) at:
32, 306464, 612896, 919328
Installing package '/altroot/cf/packages/install-tmp/junos-19.4R1.3' ...
Verified junos-boot-srxsme-19.4R1.3.tgz signed by PackageProductionEc_2019 method
ECDSA256+SHA256
Verified junos-srxsme-19.4R1.3-domestic signed by PackageProductionEc_2019 method
ECDSA256+SHA256
Verified junos-boot-srxsme-19.4R1.3.tgz signed by PackageProductionEc_2019 method
ECDSA256+SHA256 V
erified junos-srxsme-19.4R1.3-domestic signed by PackageProductionEc_2019 method
ECDSA256+SHA256
JUNOS 19.4R1.3 will become active at next reboot
WARNING: A reboot is required to load this software correctly
WARNING: Use the 'request system reboot' command
WARNING: when software installation is complete Saving state for rollback ...

```

- Use the request system software add /var/tmp/xxx command to check whether the Junos OS package is compatible for the installation with the veriexec loader.

```

username@hostname> request system software add /var/tmp/junos-srxsme-19.4R2.3.tgz
WARNING: Package junos-19.4R2.3 version 19.4R2.3 is not compatible with this system.
WARNING: Please install a package with veloadr support, 20.3 or higher.

```

In this example, you see that the installation is terminated because the veriexec loader is not supported for Junos OS Releases before 20.3R1.

Generate OpenPGP Keys for Secure Package Signing on Junos OS

SUMMARY

Learn how to use OpenPGP for secure package signing on Junos OS to ensure the authenticity and integrity of software packages.

IN THIS SECTION

- [OpenPGP Overview | 269](#)
- [Veriexec Feature for OpenPGP | 269](#)

OpenPGP Overview

OpenPGP is a widely adopted encryption standard that uses asymmetric cryptography to provide confidentiality, integrity, and non-repudiation of data. You can ensure the secure addition of customer packages by using OpenPGP. This standard facilitates self-signing and verification of the software or firmware files involved in the installation or update process on Juniper Networks devices.

You must set up a signer machine that includes a GNU Privacy Guard (GPG) installation to generate a private OpenPGP key and sign any software package.

You must keep the newly created OpenPGP key secure. Anyone who has access to the private key can sign code that can run on the customer device.

Veriexec Feature for OpenPGP

The "[Veriexec](#)" on [page 262](#) feature on Junos OS enforces that all software is signed by a valid authority, which is usually Juniper Networks, Inc. Customers can sign their own packages with OpenPGP (Pretty Good Privacy) to allow the package to run on Junos OS.

This section provides a step-by-step walk-through on how to use OpenPGP for veriexec, covering key generation, key installation on Junos OS devices, signing manifests, and verifying signed manifests using OpenPGP signatures.

Before generating OpenPGP keys, add the **veriexec-openpgp.tgz extension** to Junos OS. To add the extension, execute the command `request system software add veriexec-openpgp.tgz` on the Junos OS device. This command ensures that the necessary components for OpenPGP integration are installed and ready for use.

After you add the veriexec-openpgp.tgz extension, proceed with generating the OpenPGP keys in the signer machine.

Steps to Generate OpenPGP Keys

To set up secure package signing on Junos OS, follow the below steps:

1. Generate OpenPGP keys:
 - a. Open a terminal or command prompt.

- b. Execute the following command to generate an RSA keypair:

```
GNUPGHOME=$PWD/.gnupg gpg --openpgp --quick-generate-key --batch --passphrase '' "keyname"
RSA
```

Along with a keypair, the command generates the key ID, which will be used later for verification.

2. Export public and private keys on the signer machine:

- a. Execute the following commands to export the public and private keys on the signer machine:

```
GNUPGHOME=$PWD/.gnupg gpg --openpgp --export --armor > ACA72B4719FD2523.pub.asc

GNUPGHOME=$PWD/.gnupg gpg --openpgp --export-secret-keys --armor > ACA72B4719FD2523.sec.asc
```

Keep the exported private key file with the .sec.asc extension secret.

- b. Replace **ACA72B4719FD2523** with the key ID obtained in step 1. These commands generate ASCII-armored versions of the public and private keys.

3. Install the public key on Junos OS:

- a. Copy the public key file (**ACA72B4719FD2523.pub.asc**) to the device.
- b. Execute the following command on the device to install the public key.

```
request system trust openpgp add ACA72B4719FD2523.pub.asc
```

This command adds the public key to the trust store used by veriexec.

4. Sign manifests:

To sign manifests, you have two options depending on your requirements.

a. Option 1: Use the signing server:

- i. Create or obtain the manifest file that needs to be signed. For example: manifest.
- ii. Execute the following command to generate an OpenPGP signature for the manifest:

```
openpgp-sign.py -a -u server:port manifest
```

- iii. Replace server:port with the actual signing server and port. This command signs the manifest using the OpenPGP and generates a signature file. For example: manifest.asc.

b. Option 2: Perform self-signing using GPG on signer machine:

- i. Create or obtain the manifest file that needs to be signed. For example: manifest.
- ii. Execute the following command to sign the manifest using OpenPGP and generate a signature file. For example: manifest.asc

```
gpg -a --detach-sign manifest
```

- iii. Replace `server:port` with the actual signing server and port. This command signs the manifest using OpenPGP and generates a signature file. For example: manifest.asc

5. Verify signed manifests:

- a. Copy the signed manifest file, for example: manifest and manifest.asc to the Junos OS device.
- b. Open a Unix shell or command prompt on the Junos OS device.
- c. Execute the following command to verify the signed manifest:

```
verifexec -C /path/to/directory /path/to/manifest
```

Replace **/path/to/directory** with the actual directory path where the manifest and associated files are located, and **/path/to/manifest** with the path to the manifest file. Verifexec will automatically verify the manifest (package) using the public key and the corresponding signature.

By following these steps, you can leverage the `verifexec-openpgp` package to control the signing keys and trust anchors used by Junos OS. Use these steps to provide additional security and customization options for customer packages.

4

PART

Install and Upgrade the BIOS and Firmware

- For Routers | 273
 - For Switches | 283
 - For Firewalls | 304
-

For Routers

IN THIS CHAPTER

- [Installing and Upgrading Firmware | 273](#)

Installing and Upgrading Firmware

SUMMARY

To get optimal network performance and to fix a vulnerability, you can upgrade the firmware on your device.

IN THIS SECTION

- [Before You Begin Installing or Upgrading the Firmware | 273](#)
- [Installing Firmware on the 100-Gigabit DWDM OTN MIC \(MIC3-100G-DWDM\) | 275](#)
- [Upgrading Firmware on the 100-Gigabit DWDM OTN MIC \(MIC3-100G-DWDM\) | 276](#)
- [Installing Firmware on ACX6360 Router | 278](#)
- [Upgrading Firmware on the ACX6360 Router | 279](#)

Before You Begin Installing or Upgrading the Firmware

Before you begin installing or upgrading the firmware on the MIC or PIC, complete the following steps:

1. Verify that a previous version of the firmware package is installed on the router by using the `show version` command. For example:

show version (MX240, MX480, MX960, MX2010, MX2020)

```

user@host> show version
Hostname: mxHost
Model: mx480
Junos: 15.1I20160816_2117_yyin
JUNOS OS Kernel 64-bit (WITNESS) [20160723.102341_fbsd-builder_stable_10]
JUNOS OS libs [20160723.102341_fbsd-builder_stable_10]
JUNOS OS runtime [20160723.102341_fbsd-builder_stable_10]
JUNOS OS time zone information [20160723.102341_fbsd-builder_stable_10]
...
JUNOS jfirmware [20160628.005233_builder_release_151_f_throttle]
JUNOS Online Documentation [20160812.205759_yyin_release_151_f_throttle]
JUNOS FIPS mode utilities [20160816.211724_yyin_release_151_f_throttle]
....

```

show version (SRX1600, SRX2300, SRX4120, SRX4300 and SRX4700)

```

user@host> show version
Hostname: YYY

Model: srx1600
Junos: 24.2I-20240219_dev_common.0.0827
JUNOS OS Kernel 64-bit [20240207.32ed862a0f7__ci_fbsd_builder_main]
JUNOS modules [20240219.195852__cd-builder_dev_common]
JUNOS srxtvp modules [20240219.195852__cd-builder_dev_common]
.....

JUNOS jfirmware [20231123.175230_builder_junos_234_r1]
JUNOS Online Documentation [20240219.195852__cd-builder_dev_common]
JUNOS FIPS mode utilities [20240219.195852__cd-builder_dev_common]
.....

```

If the output of the `show version` command displays `JUNOS jfirmware` among the list of packages that are installed on the router, then a previous version of the firmware package is installed on the router. If the output of the `show version` command does not display `JUNOS jfirmware` among the list of packages that are installed on the router, a firmware package is not installed on the router.

2. If a previous version of the firmware package is installed on the router, delete the firmware package from the router by using the `request system software delete jfirmware` command.

```
user@host> request system software delete jfirmware
/packages/db/jfirmware-x86-32-15.1F-20160625.0
```

3. To verify that the firmware package is removed from the router, use the `show version` command.

```
user@host> show version
Hostname: mxHost
Model: mx240
Junos: 15.1F6-S1.3
JUNOS OS Kernel 64-bit [20160724.331042_builder_stable_10]
JUNOS OS libs [20160724.331042_builder_stable_10]
JUNOS OS runtime [20160724.331042_builder_stable_10]
JUNOS OS time zone information [20160724.331042_builder_stable_10]
....
JUNOS IDP Services [20160812.205945_builder_junos_151_f6_s1]
....
JUNOS Packet Forwarding Engine Support (M/T Common) [20160812.205945_builder_junos_151_f6_s1]
JUNOS Online Documentation [20160812.205945_builder_junos_151_f6_s1]
JUNOS FIPS mode utilities [20160812.205945_builder_junos_151_f6_s1]
```

If the firmware package is uninstalled successfully, the output of the `show version` command does not display JUNOS `jfirmware` among the list of packages that are installed on the router.

Installing Firmware on the 100-Gigabit DWDM OTN MIC (MIC3-100G-DWDM)

Before you install the firmware package, ensure that a previous version is not installed on the router. For more information, see ["Before You Begin Installing or Upgrading the Firmware" on page 273](#).

To install the firmware package, complete the following steps:

1. Upgrade Junos OS on the router to the version that supports the firmware package. See [Installing the Software Package on a Device with Redundant Routing Engines \(Junos OS\)](#) or ["Installing the Software Package on a Router with a Single Routing Engine \(Junos OS\)" on page 102](#) for more information.
2. Download the firmware package from <https://support.juniper.net/support/downloads/>. For information about downloading software packages, see ["Downloading Software \(Junos OS\)" on page 73](#).



NOTE: Download the firmware package specific to your router. The firmware package for MX Series routers is different from the firmware package for the PTX Series routers.

3. Save the firmware package to the `/var/path/package-name` directory on the router. For example, you can save the firmware package to the `/var/tmp` directory.
4. Install the firmware package by using the `request system software add /var/path/package-name` command. For example, to install the `jfirmware-x86-32-15.1F6.9.tgz` package:

```
user@host> request system software add jfirmware-x86-32-15.1F6.9.tgz
```

5. Run the `show version` command to verify that the firmware package is installed.

```
user@host> show version
Hostname: Host1
Model: mx480
Junos: 15.1I20160816_2117_yyin
JUNOS OS Kernel 64-bit (WITNESS) [20160723.102341_fbsd-builder_stable_10]
JUNOS OS libs [20160723.102341_fbsd-builder_stable_10]
JUNOS OS runtime [20160723.102341_fbsd-builder_stable_10]
JUNOS OS time zone information [20160723.102341_fbsd-builder_stable_10]
...
JUNOS jfirmware [20160628.005233_builder_release_151_f_throttle]
JUNOS Online Documentation [20160812.205759_yyin_release_151_f_throttle]
JUNOS FIPS mode utilities [20160816.211724_yyin_release_151_f_throttle]
....
```

After the firmware package is installed successfully, the output of the `show version` command displays `JUNOS jfirmware..` among the list of packages that are installed on the router.

Upgrading Firmware on the 100-Gigabit DWDM OTN MIC (MIC3-100G-DWDM)

Before you upgrade the firmware package, ensure that a previous version is not installed on the router. For more information, see ["Before You Begin Installing or Upgrading the Firmware" on page 273](#).

To upgrade the version of your firmware package, complete the following steps:

1. Run the `show system firmware` command to view the list of components installed on the router and the firmware version for each component.

```
user@host> show system firmware
```

Part	Type	Tag	Current	Available	Status
------	------	-----	---------	-----------	--------

		version	version		
FPC 0	ROM Monitor	0 0	10.4.1		OK
FPC 1	ROM Monitor	0 0	10.4.1		OK
FPC 2	ROM Monitor	0 0	10.4.1		OK
PIC 0	CMIC LTC 2/0	1	.0	1.0	OK
FPC 3	ROM Monitor	0 0	10.4.1		OK
FPC 4	ROM Monitor	0 0	13.3.1		OK
FPC 4	MPCS(0)	2	0.24.0		OK
Routing Engine 0	RE BIOS	0	1.18		OK
Routing Engine 1		0	1.18		OK

The output of the `show system firmware` command displays the current firmware version of the MIC as .0 and the available firmware version as 1.0.

- To upgrade the firmware of the MIC, use the `request system firmware upgrade pic` command. For example, to upgrade the firmware version of the MIC from .0 to 1.0, specify the MPC slot and MIC slot in the command.

```
user@host> request system firmware upgrade pic pic-slot 0 fpc-slot 2
```

Part	Type	Tag	Current	Available	Status
		version	version		
FPC 2					
PIC 0	CMIC LTC 2/0	1	.0	1.0	OK

Perform indicated firmware upgrade ? [yes,no] (no) yes

Confirm that you want to perform the firmware upgrade by typing Yes so the firmware upgrade is initiated.

- To monitor the progress of the upgrade, use the `show system firmware` command. During the installation process, the status of the MIC changes to `PROGRAMMING`. When the installation process is complete, the status of the MIC changes to `UPGRADED SUCCESSFULLY`.



NOTE: The amount of time it takes to upgrade firmware varies depending on the component.

```
user@host> show system firmware
```

Part	Type	Tag	Current	Available	Status
		version	version		
FPC 0	ROM Monitor	0 0	10.4.1		OK
FPC 1	ROM Monitor	0 0	10.4.1		OK
FPC 2	ROM Monitor	0 0	10.4.1		OK

PIC 0	CMIC LTC 2/0	1	.0	1.0	OK
FPC 3	ROM Monitor 0	0	10.4.1		OK
FPC 4	ROM Monitor 0	0	13.3.1		OK
FPC 4	MPCS(0)	2	0.24.0		OK
Routing Engine 0	RE BIOS	0	1.18		OK
Routing Engine 1		0	1.18		OK



NOTE: If the installation process fails, delete the firmware package by using the request `system software delete firmware-package-name` command. Reinstall the firmware package by following the procedure for installing the firmware package and then upgrade the firmware package.

- Restart the MPC that the MIC is installed in by using the request `chassis fpc fpc-slot restart` command.
- (Optional) After the firmware upgrade is successfully completed, uninstall the firmware package from the router by using the request `system software delete` command.

Installing Firmware on ACX6360 Router

Before you install the firmware package, ensure that a previous version is not installed on the router. For more information, see ["Before You Begin Installing or Upgrading the Firmware" on page 273](#).

To install the firmware package, complete the following steps:

- Upgrade Junos OS on the router to the version that supports the firmware package. See [Installing the Software Package on a Device with Redundant Routing Engines \(Junos OS\)](#) or ["Installing the Software Package on a Router with a Single Routing Engine \(Junos OS\)" on page 102](#) for more information.
- Download the firmware package from <https://support.juniper.net/support/downloads/>. For information about downloading software packages, see ["Downloading Software \(Junos OS\)" on page 73](#).



NOTE: Download the firmware package specific to your router. The firmware package for ACX Series routers is different from the firmware package for the MX or PTX Series routers.

- Save the firmware package to the `/var/path/package-name` directory on the router. For example, you can save the firmware package to the `/var/tmp` directory.

4. Install the firmware package by using the `request system software add /var/path/package-name` command. For example, to install the `jfirmware-x86-32-15.1F6.9.tgz` package:

```
user@host> request system software add jfirmware-x86-32-15.1F6.9.tgz
```

5. Run the `show version` command to verify that the firmware package is installed.

```
user@host> show version
Hostname: YYY

Model: ACX6360-OR
Junos: 18.3I20180430_1917_XXX
JUNOS OS Kernel 64-bit (WITNESS) [20180413.173511_fbsd-builder_stable_11]
JUNOS OS libs [20180413.173511_fbsd-builder_stable_11]
JUNOS OS runtime [20180413.173511_fbsd-builder_stable_11]
JUNOS OS time zone information [20180413.173511_fbsd-builder_stable_11]
...
JUNOS jfirmware [20180430.191738_XXX_dev_common]
JUNOS Online Documentation [20180430.191738_XXX_dev_common]
JUNOS jail runtime [20180413.173511_fbsd-builder_stable_11]
....
```

After the firmware package is installed successfully, the output of the `show version` command displays `JUNOS jfirmware..` among the list of packages that are installed on the router.

Upgrading Firmware on the ACX6360 Router

Before you upgrade the firmware package, ensure that a previous version is not installed on the router. For more information, see ["Before You Begin Installing or Upgrading the Firmware" on page 273](#).

To upgrade the version of your firmware package, complete the following steps:

1. Run the `show system firmware` command to view the list of components installed on the router and the firmware version for each component.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Pseudo CB 0	CB FPGA	0	2.12.0	2.12.0	OK
Pseudo CB 0	PORT FPGA	9	1.14.0	1.15.0	OK
Pseudo CB 0	TIC FPGA	11	4101.5.0	4101.5.0	OK

FPC 0		0	0.0.0	71.63d	OK
PIC 1	DWDM DCO-0/1/0	20	38.1.9	38.2.6	OK
PIC 1	DWDM DCO-0/1/1	21	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/2	22	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/3	23	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/4	24	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/5	25	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/6	26	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/7	27	1.0.0	38.2.6	OK
Routing Engine 0	RE BIOS	7	0.24.1	0.24.01	OK
Routing Engine 0	RE FPGA	2	9.6.0	9.9.0	OK
Routing Engine 0	RE SSD1	3	12028		OK
Routing Engine 0	RE SSD2	4	12028		OK
Power Supply 0		0	0.0.0		OK
Power Supply 1		0	0.0.0		OK

The output of the `show system firmware` command displays the current firmware version of the PIC as .0 and the available firmware version as 1.0.

2. To upgrade the firmware of the PIC, for ACX6360 use the `request system firmware upgrade pic` command. For example, to upgrade the firmware version of the PIC from .0 to 1.0, specify the FPC slot and PIC slot in the command.

```
user@host> request system firmware upgrade pic fpc-slot 0 pic-slot 1
```

Part	Type	Tag	Current version	Available version	Status
FPC 0					
PIC 1	DWDM DCO-0/1/0	20	38.2.9	38.2.6	OK

Perform indicated firmware upgrade ? [yes,no] (no) **yes**

Confirm that you want to perform the firmware upgrade by typing Yes so the firmware upgrade is initiated.

3. To monitor the progress of the upgrade, use the `show system firmware` command. During the installation process, the status of the PIC changes to `PROGRAMMING`. When the installation process is complete, the status of the PIC changes to `UPGRADED SUCCESSFULLY`.



NOTE: The amount of time it takes to upgrade firmware varies depending on the component.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Pseudo CB 0	CB FPGA	0	2.12.0	2.12.0	OK
Pseudo CB 0	PORT FPGA	9	1.14.0	1.15.0	OK
Pseudo CB 0	TIC FPGA	11	4101.5.0	4101.5.0	OK
FPC 0		0	0.0.0	71.63d	OK
PIC 1	DWDM DCO-0/1/0	20	38.2.6	38.2.6	OK
PIC 1	DWDM DCO-0/1/1	21	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/2	22	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/3	23	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/4	24	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/5	25	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/6	26	1.0.0	38.2.6	OK
PIC 1	DWDM DCO-0/1/7	27	1.0.0	38.2.6	OK
Routing Engine 0	RE BIOS	7	0.24.1	0.24.01	OK
Routing Engine 0	RE FPGA	2	9.6.0	9.9.0	OK
Routing Engine 0	RE SSD1	3	12028		OK
Routing Engine 0	RE SSD2	4	12028		OK
Power Supply 0		0	0.0.0		OK
Power Supply 1		0	0.0.0		OK



NOTE: If the installation process fails, delete the firmware package by using the request `system software delete firmware-package-name` command. Reinstall the firmware package by following the procedure for installing the firmware package and then upgrade the firmware package.

- Restart the ACX6360 router by using request `chassis fpc restart slot 0` command for the upgrade to take effect.
- (Optional) After the firmware upgrade is successfully completed, uninstall the firmware package from the router by using the request `system software delete` command.

RELATED DOCUMENTATION

| [show system firmware](#)

For Switches

IN THIS CHAPTER

- [Upgrading System CPLD, BIOS, CPU CPLD, PoE Firmware, and eMMC Firmware for EX4400 Devices | 283](#)
- [Upgrading U-Boot, PoE Firmware, eUSB Firmware, and System CPLD for EX4100 Devices | 295](#)
- [Upgrade Firmware on QFX5120-48Y and EX4650-48Y using the jfirmware Package | 300](#)

Upgrading System CPLD, BIOS, CPU CPLD, PoE Firmware, and eMMC Firmware for EX4400 Devices

SUMMARY

The following sections describe the steps to upgrade BIOS, CPU CPLD, System CPLD, and Virtual Chassis (VC) firmware in EX4400.

IN THIS SECTION

- [Checking Latest Firmware Versions | 283](#)
- [Upgrading BIOS | 284](#)
- [Upgrading CPU CPLD | 285](#)
- [Upgrading System CPLD | 286](#)
- [Upgrading PoE Firmware | 288](#)
- [Upgrading PoE Firmware Using jfirmware | 290](#)
- [Upgrading eMMC Firmware | 293](#)
- [Upgrading Firmware in Virtual Chassis | 294](#)

Checking Latest Firmware Versions

To check the latest firmware versions available, execute the `show system firmware` command. If a new firmware version is available, upgrade using the latest firmware package.



NOTE: You can download the firmware package from <https://support.juniper.net/support/downloads/?p=ex4400>.

Upgrading BIOS

Perform the following steps to upgrade the BIOS:

1. Add jfirmware.

```
user@host> request system software add /var/tmp/jfirmware-ex.tgz
```

2. Upgrade BIOS.

```
user@host> request system firmware upgrade jfirmware bios
```

3. Check the progress of the BIOS upgrade.

```
user@host> request system firmware upgrade jfirmware bios progress
```

4. Reboot the device for the upgrade to take effect.

```
root>request system reboot
```

5. Use the following command to confirm the BIOS version.

Before upgrade:

```
user@host> show chassis firmware
Part           Type      Version
FPC 0          loader    FreeBSD EFI loader 2.0
               BIOS      CDEN_P_EX1_00.15.01.00
               System CPLD 0.f
               CPU CPLD   1.0
```

After upgrade:

```
user@host> show chassis firmware
Part           Type      Version
```


2. Check the availability of a new PoE version in the latest Junos version.

```
user@host> show poe controller
```

```
user@host> show poe controller
Controller  Maximum   Power      Guard    Management  Status      Lldp
index      power     consumption band
0**        740W      0.00W      0W       Class       BT_MODE     Disabled

**New PoE software upgrade available.
Use 'request system firmware upgrade poe fpc-slot <slot>'
This procedure will take around 10 minutes (recommended to be performed during maintenance)
```

3. Upgrade the PoE firmware.

```
request system firmware upgrade poe fpc-slot <slot>
```

```
Firmware upgrade initiated. Poe Upgrade takes about 10 minutes
Use 'show poe controller' to get the download status,
Please Reboot the system after Upgrade is complete.
```

4. Monitor the PoE upgrade under the "status" field.

```
user@host> show poe controller | refresh 60
```

```
user@host> Controller  Maximum   Power      Guard    Management  Status      Lldp
index      power     consumption band
0**        740W      0.00W      0W       SW_DOWNLOAD(5%) Disabled

**New PoE software upgrade available.
Use 'request system firmware upgrade poe fpc-slot <slot>'
This procedure will take around 10 minutes (recommended to be performed during maintenance)
---(refreshed at 2022-05-11 06:58:49 UTC)---
---(refreshed at 2022-05-11 07:06:49 UTC)---
Controller  Maximum   Power      Guard    Management  Status      Lldp
index      power     consumption band
0**        740W      0.00W      0W       SW_DOWNLOAD(100%) Disabled

**New PoE software upgrade available.
Use 'request system firmware upgrade poe fpc-slot <slot>'
```

This procedure will take around 10 minutes (recommended to be performed during maintenance)
 ---(refreshed at 2022-05-11 07:07:49 UTC)---

Controller index	Maximum power	Power consumption	Guard band	Management	Status	Lldp Priority
0	740W	0.00W	0W		BT_MODE	Disabled

5. (Recommended) Perform a system halt using the following command.

```
user@host> request system halt
```

6. Power cycle the device for the upgrade to take effect.

7. Check the upgraded PoE.

Before upgrade:

```
user@host> show chassis firmware detail
FPC 0
  PoE firmware          1.3.0.9.0
  Boot Firmware
    U-Boot              1.0
  Boot Firmware
    loader              FreeBSD EFI loader 2.0
```

After upgrade:

```
user@host> show chassis firmware detail
FPC 0
  PoE firmware          1.3.0.11.0
  Boot Firmware
    U-Boot              1.0
  Boot Firmware
    loader              FreeBSD EFI loader 2.0
```

Upgrading PoE Firmware Using jfirmware

1. Check the current firmware version.

```
user@host> show chassis firmware detail
FPC 0
  PoE firmware          1.3.0.9.0
  Boot Firmware
```

U-Boot	1.0
Boot Firmware loader	FreeBSD EFI loader 2.0

2. Check the availability of a new PoE version in the latest Junos version.

```
user@host> show poe controller
```

```
user@host> show poe controller
```

Controller index	Maximum power	Power consumption	Guard band	Management Class	Status	Lldp Priority
0**	740W	0.00W	0W	Class	BT_MODE	Disabled

****New PoE software upgrade available.**

Use 'request system firmware upgrade poe fpc-slot <slot>'

This procedure will take around 10 minutes **(recommended to be performed during maintenance)**

3. Upgrade the PoE firmware using jfirmware.

```
request system firmware upgrade jfirmware poe file jfirmware-file fpc-slot (number | all-members)
<poe-at-firmware | poe-bt-firmware>
```

- file (jfirmware-file) is the file name of the jfirmware package.
- fpc-slot (number | all-members) is the Virtual Chassis member or line card specified by number.

For example:

```
user@host> request system firmware upgrade jfirmware poe file jfirmware-ex.tgz fpc-slot 0
```

Firmware upgrade initiated. Poe Upgrade takes about 10 minutes

Use 'show poe controller' to get the download status,

Please Reboot the system after Upgrade is complete.

4. Monitor the PoE upgrade under the "status" field.

```
user@host> show poe controller | refresh 60
```

```
user@host> Controller Maximum Power Guard Management Status Lldp
index power consumption band Priority
0** 740W 0.00W 0W SW_DOWNLOAD(5%) Disabled
**New PoE software upgrade available.
Use 'request system firmware upgrade poe fpc-slot <slot>'
This procedure will take around 10 minutes (recommended to be performed during maintenance)
---(refreshed at 2022-05-11 06:58:49 UTC)---
---(refreshed at 2022-05-11 07:06:49 UTC)---
Controller Maximum Power Guard Management Status Lldp
index power consumption band Priority
0** 740W 0.00W 0W SW_DOWNLOAD(100%) Disabled
**New PoE software upgrade available.
Use 'request system firmware upgrade poe fpc-slot <slot>'
This procedure will take around 10 minutes (recommended to be performed during maintenance)
---(refreshed at 2022-05-11 07:07:49 UTC)---
Controller Maximum Power Guard Management Status Lldp
index power consumption band Priority
0 740W 0.00W 0W BT_MODE Disabled
```

5. (Recommended) Perform a system halt using the following command.

```
user@host> request system halt
```

6. Power cycle the device for the upgrade to take effect.

7. Check the upgraded PoE.

```
user@host> show chassis firmware detail
```

```
user@host> show chassis firmware detail
FPC 0
PoE firmware 1.3.0.11.0
Boot Firmware
U-Boot 1.0
```

Boot Firmware	
loader	FreeBSD EFI loader 2.0

Upgrading eMMC Firmware

Perform the following steps to upgrade the eMMC firmware:

1. Add jfirmware.

```
user@root> request system software add /var/tmp/jfirmware-ex.tgz
```

2. Upgrade eMMC firmware.

```
user@root> request system firmware upgrade jfirmware mmc
```

3. Check the progress of the upgrade.

```
user@root> request system firmware upgrade jfirmware mmc progress
```

4. Reboot the system.

```
user@root> request system reboot
Reboot the system ? [yes,no] (no) yes
```

5. Use the following command to confirm the eMMC version.

Before upgrade:

```
user@host> show system storage mmc status mmcscd0

Showing MMC status information
Device : mmcscd0
  General information
  -----
  Disk size           : 20635975680B (19GB)
  Product name        : ATPBG2
  Product revision     : 1.3
  Product serial number : 894418974 (0x354FC01E)
  Manufacturing Date   : 05/2021
  Manufacturer         : Unrecognized
  Firmware version      : Q92-6192817UJ05P33      Health status
```

```

-----
Pre EOL Information      : Normal
Life time estimate Type-A : 0% - 10% device life time used

```

After upgrade:

```

user@host> show system storage mmc status mmc0

Device : mmc0
General information
-----
Disk size           : 20635975680B (19GB)
Product name        : ATPBG2
Product revision     : 1.3
Product serial number : 894418974 (0x354FC01E)
Manufacturing Date   : 05/2021
Manufacturer         : Unrecognized
Firmware version      : R92-6192817TH12P55      Health status  <<<<<<<
-----
Pre EOL Information      : Normal
Life time estimate Type-A : 0% - 10% device life time used

```

Upgrading Firmware in Virtual Chassis

Perform the following steps to upgrade the VC firmware:

1. Add the jfirmware package in primary Routing Engine (RE).

```

user@host> request system software add /var/tmp/jfirmware-ex.tgz

```

2. Upgrade the BIOS and check the status in primary RE.

```

user@host> request system firmware upgrade jfirmware bios

```

```

user@host> request system firmware upgrade jfirmware bios progress

```

3. Upgrade the System CPLD and check the status in primary RE.

```
user@host> request system firmware upgrade jfirmware cpld sys
```

```
user@host> request system firmware upgrade jfirmware cpld sys progress
```

4. Upgrade the CPU CPLD and check the status in primary RE.

```
user@host> request system firmware upgrade jfirmware cpld cpu
```

```
user@host> request system firmware upgrade jfirmware cpld cpu progress
```

5. Repeat the steps 2, 3, and 4 in each member of the VC by logging in to the member. Use the request session member <fpc-slot> command to log in.
6. Power cycle or reboot all the members of the VC. Reboot the members one by one after the upgrade is successful after Step 4 depending on either CPLD or BIOS upgrade.

Upgrading U-Boot, PoE Firmware, eUSB Firmware, and System CPLD for EX4100 Devices

SUMMARY

The following sections describe the steps to upgrade U-boot, CPLD, and VC firmware in EX4100.

IN THIS SECTION

- [Upgrading U-Boot | 295](#)
- [Upgrading System CPLD | 297](#)
- [Upgrading PoE Firmware | 298](#)
- [Upgrading PoE Firmware Using jfirmware | 298](#)
- [Upgrading eUSB Firmware | 298](#)
- [Upgrading Firmware in Virtual Chassis | 299](#)

Upgrading U-Boot

PSUs	5.5-5.5
PMIC	R0A

Upgrading System CPLD

Perform the following steps to upgrade the system CPLD:

1. Add jfirmware.

```
user@host> request system software add /var/tmp/jfirmware-ex.tgz
```

2. Upgrade system CPLD.

```
user@host> request system firmware upgrade jfirmware cpld sys
```

3. Check the progress of the CPLD upgrade.

```
user@host> request system firmware upgrade jfirmware cpld sys progress
```

4. (Recommended) Perform a system halt using the following command.

```
user@host> request system halt
```

5. Power cycle the device for the upgrade to take effect.
6. Use the following commands to confirm the CPLD version.

Before upgrade:

```
user@root> show system firmware
```

Part	Type	Tag Current version	Available version	Status
FPC 0	CPLD	1.46	1.4a	OK
FPC 0	U-Boot	1.2	1.6	OK

After upgrade:

```
user@host> show chassis firmware
```

Part	Type	Version
FPC 0	U-Boot	1.6
	loader	FreeBSD EFI loader 2.0

CPLD	1.4a
eUSB	200403
PSUs	5.5-5.5
PMIC	R0A

Upgrading PoE Firmware

See ["Upgrading PoE Firmware" on page 288](#).

Upgrading PoE Firmware Using jfirmware

["Upgrading PoE Firmware Using jfirmware" on page 290](#)

See

Upgrading eUSB Firmware

Perform the following steps to upgrade the eUSB firmware:

1. Add jfirmware.

```
user@root> request system software add /var/tmp/jfirmware-ex.tgz
```

2. Upgrade the eUSB firmware.

```
user@root> request system firmware upgrade jfirmware eusb
```

Initiates the firmware upgrade.

3. Check the progress of the upgrade.

```
user@root> request system firmware upgrade jfirmware eusb progress
```

4. Use the following commands to confirm the eUSB version.

Before upgrade:

```
user@host> show chassis firmware
Part          Type      Version
FPC 0         U-Boot    1.6
               loader    FreeBSD EFI loader 2.0
               CPLD      1.48
```


4. Repeat steps 2 and 3 in each member of the VC by logging in to the member. Use the request session member <fpc-slot> to log in.
5. Power cycle or reboot all the members of the VC. Reboot the members one by one after the upgrade is successful after Step 4 depending on either CPLD or BIOS upgrade.

Upgrade Firmware on QFX5120-48Y and EX4650-48Y using the jfirmware Package

SUMMARY

This topic provides the steps to upgrade firmware on QFX5120-48Y and EX4650-48Y using the jfirmware package.

1. Check the current firmware version.

```
user@host> show chassis firmware
```

Part	Type	Version
FPC 0	U-Boot	***
	loader	FreeBSD/i386 bootstrap loader 1.2
	BIOS	CBDE_P_LIG0_00.25.1
	TMC FPGA	7.2.0
	Optics CPLD	0.1.4
	RE FPGA	0.3.3
	Power CPLD	2.0

2. Check the availability of a new jfirmware version in the [Junos download](#) page and download the package.
3. Upgrade the TMC firmware using jfirmware.

```
user@host> request system firmware upgrade jfirmware fpga tmc file jfirmware-file
```



NOTE: *jfirmware-file* is the file name of the jfirmware package.

```
Validated qfx-5e-jfirmware-18.3R1.10.tgz
TMC_FPGA firmware upgrade initiated
Check progress using "request system firmware upgrade jfirmware fpga <type> progress"
```

4. Check the progress of the upgrade.

```
user@host> request system firmware upgrade jfirmware fpga tmc progress
TMC_FPGA upgrade successfull
Please power cycle the device to complete the upgrade
```



NOTE: Proceed to next step for the next firmware component upgrade only after the current upgrade is complete.

5. Upgrade the RE firmware using jfirmware.

```
user@host> request system firmware upgrade jfirmware fpga re file jfirmware-file
```

```
Validated qfx-5e-jfirmware-18.3R1.10.tgz
RE_FPGA firmware upgrade initiated
Check progress using "request system firmware upgrade jfirmware fpga <type> progress"
```

6. Check the progress of the upgrade.

```
user@host> request system firmware upgrade jfirmware fpga re progress
RE_FPGA upgrade successfull
Please power cycle the device to complete the upgrade
```



NOTE: Proceed to next step for the next firmware component upgrade only after the current upgrade is complete.

7. Upgrade the powercpld firmware using jfirmware.

```
user@host> request system firmware upgrade jfirmware fpga powercpld file jfirmware-file
```

```
Validated qfx-5e-jfirmware-18.3R1.10.tgz
PWR_CPLD firmware upgrade initiated
Check progress using "request system firmware upgrade jfirmware fpga <type> progress"
```

8. Check the progress of the upgrade.

```
user@host> request system firmware upgrade jfirmware fpga powercpld progress
PWR_CPLD upgrade successfull
Please power cycle the device to complete the upgrade
```



NOTE: Proceed to next step for the next firmware component upgrade only after the current upgrade is complete.

9. Upgrade the optics_qsfp firmware using jfirmware.

```
user@host> request system firmware upgrade jfirmware fpga qsfp file jfirmware-file
```

```
Validated qfx-5e-jfirmware-18.3R1.10.tgz
OPTICS_QSFP firmware upgrade initiated
Check progress using "request system firmware upgrade jfirmware fpga <type> progress"
```

10. Check the progress of the upgrade.

```
user@host> request system firmware upgrade jfirmware fpga qsfp progress
OPTICS_QSFP upgrade successfull
Please power cycle the device to complete the upgrade
```



NOTE: Proceed to next step for the next firmware component upgrade only after the current upgrade is complete.

11. Upgrade the optics_sfp firmware using jfirmware.

```
user@host> request system firmware upgrade jfirmware fpga sfp file jfirmware-file
```

```
Validated qfx-5e-jfirmware-18.3R1.10.tgz
OPTICS_SFP firmware upgrade initiated
Check progress using "request system firmware upgrade jfirmware fpga <type> progress"
```

12. Check the progress of the upgrade.

```
user@host> request system firmware upgrade jfirmware fpga sfp progress
OPTICS_SFP upgrade successfull
Please power cycle the device to complete the upgrade
```



NOTE: Proceed to next step only after the current upgrade is complete.

13. Power cycle the device for the upgrade to take effect.
14. Check the upgraded firmware.

```
user@host> show chassis firmware detail
```

For Firewalls

IN THIS CHAPTER

- [Upgrade BIOS and Firmware \(SRX Series Firewall Only\) | 304](#)
- [Upgrade 5.1KW HVAC/HVDC Single and Dual Input PSM Firmware \(SRX5800\) | 307](#)

Upgrade BIOS and Firmware (SRX Series Firewall Only)

SUMMARY

Learn how to upgrade BIOS and Firmware for SRX Series Firewall.

IN THIS SECTION

- [BIOS Upgrades on SRX Series Firewalls | 304](#)
- [Disable Auto BIOS Upgrade on SRX Series Firewalls | 306](#)

BIOS Upgrades on SRX Series Firewalls

IN THIS SECTION

- [Manual BIOS Upgrade Using the Junos CLI | 304](#)
- [Auto BIOS Upgrade Methods on SRX Series Firewalls | 305](#)

Manual BIOS Upgrade Using the Junos CLI

On SRX300 Line of Series Firewall devices, the BIOS consists of a U-boot and the Junos loader. Additionally, a backup BIOS is supported which includes a backup copy of the U-boot in addition to the active copy from which the system generally boots up.

Table 18 on page 305 Lists the CLI commands used for manual BIOS upgrade.

Table 18: CLI Commands for Manual BIOS Upgrade

Active BIOS	Backup BIOS
<code>request system firmware upgrade re bios</code>	<code>request system firmware upgrade re bios backup</code>

BIOS upgrade procedure:

1. Install the jloader-srxsme package.

- a. Copy the jloader-srxsme signed package to the device. The version of the jloader-srxsme package you install must match the version of Junos OS.
- b. Install the package using the `request system software add <path to jloader-srxsme package> no-copy no-validate` command.

Installing the jloader-srxsme package places the necessary images under `directory/boot`.

2. Verify that the required images for upgrade are installed. Use the `show system firmware` to verify that the correct BIOS image version is available for upgrade.

3. Upgrade the BIOS (Active and backup) image.

Active BIOS:

- a. Initiate the upgrade using the `request system firmware upgrade re bios` command.
- b. Monitor the upgrade status using the `show system firmware` command.

The device must be rebooted for the upgraded active BIOS to take effect.

Backup BIOS:

- a. Initiate the upgrade using the `request system firmware upgrade re bios backup` command.
- b. Monitor the upgrade status using the `show system firmware` command.

Auto BIOS Upgrade Methods on SRX Series Firewalls

The BIOS version listed in the **bios-autoupgrade.conf** file is the minimum supported version. If the current device has a BIOS version earlier than the minimum compatible version, then the auto BIOS upgrade feature upgrades the BIOS automatically to the latest version.

The BIOS upgrades automatically in the following scenarios:

- During Junos OS, upgrade either through the J-Web user interface or the CLI (using the `request system software add no-copy no-validate software-image`). In this case, only the active BIOS is upgraded.
- During loader installation using TFTP or USB (using the `install tftp:///software-image` command). In this case, only the active BIOS is upgraded.
- During system boot-up. In this case, both the active BIOS and the backup BIOS are upgraded.

Disable Auto BIOS Upgrade on SRX Series Firewalls

The auto BIOS upgrade feature is enabled by default. You can disable the feature using the CLI in configuration mode.

To disable the automatic upgrade of the BIOS on an SRX Series Firewall, use the `chassis routing-engine bios` command as following:

```
user@host# set chassis routing-engine bios no-auto-upgrade
```

The command disables automatic upgrade of the BIOS only during Junos OS upgrade or a system boot-up. It does not disable automatic BIOS upgrade during loader installation.

The `set chassis routing-engine bios uninterrupt` command is introduced on SRX300, SRX320, SRX340, and SRX345 devices to disable user inputs at U-Boot and boot loader stage. The `set chassis routing-engine bios uninterrupt` command is introduced for SRX380 Series devices.

The `set chassis routing-engine bios uninterrupt` is available on vSRX3.0 devices.

The `set chassis routing-engine bios uninterrupt` command can be used on SRX300, SRX320, SRX340, and SRX345, devices to disable user inputs at U-Boot, boot loader and Junos-Kernel boot stage. The `set chassis routing-engine bios uninterrupt` command is introduced in Junos OS on SRX380 Series devices.

To disable the user inputs at U-Boot, boot loader and Junos Kernel boot stage, use the `chassis routing-engine bios` command as following:

```
user@host# set chassis routing-engine bios uninterrupt
```

Install U-Boot version 3.2 or later and loader version 2.9 or later on SRX Series Firewalls to disable user inputs at the U-Boot and boot loader stage with the `chassis routing-engine bios` command.

You can check the version number at console output when your device boots up as shown in the following sample:

```
U-Boot 2013.07-JNPR-3.4 (Build time: Aug 02 2017 - 18:57:37)
FreeBSD/MIPS U-Boot bootstrap loader, Revision 2.9
```

You can also check the U-Boot and loader version at Junos shell prompt as shown the following sample:

```
root@% kenv
  LINES="24"
  boot.ver="3.5"
  loader.name="FreeBSD/MIPS U-Boot bootstrap loader"
  loader.version="2.9"
root@%
```

On SRX Series Firewalls, if both set system ports console insecure and set chassis routing-engine bios uninterrupt options are configured, no alternative recovery method is available if the Junos OS fails to boot and the device becomes unusable.

RELATED DOCUMENTATION

| [Install Software on SRX Series Devices](#)

Upgrade 5.1KW HVAC/HVDC Single and Dual Input PSM Firmware (SRX5800)

SUMMARY

Learn how to upgrade 5.1KW HVAC/HVDC Single and Dual Input PSM Firmware on SRX5800 device.

The 5.1KW HVAC/HVDC Single and Dual Input PSM on SRX5800 provides a maximum output power of 5100W and supports AC or DC input. In single feed mode, the PSM provides power at a reduced

capacity of 2550W. In dual feed mode, the PSM provides power at a full capacity of 5100W. The PSM supports a 1+1 redundancy. To upgrade firmware for SRX5800 PSM:

1. Disable alarms related to the PSMs in the system using `show chassis alarms` command.
2. The device chassis must have redundant PSMs. Power feeds (two, in case of dual input) of the PSMs must be upgraded, connected, and energized.
3. The AC/DC input must show status OK (2 active and connected feeds).
4. Perform the firmware upgrade one PSM at a time. Also, do not run any test or CLI commands after the upgrade is started using `request system firmware upgrade pem slot` . Use `show system firmware` command to check the upgrade status.

RELATED DOCUMENTATION

show chassis power

show chassis firmware

show chassis hardware

5

PART

System Backup and Recovery

- [Back Up an Installation | 310](#)
 - [Recover Junos OS | 328](#)
-

Back Up an Installation

IN THIS CHAPTER

- [Backing Up an Installation Using Snapshots \(Junos OS\) | 310](#)

Backing Up an Installation Using Snapshots (Junos OS)

IN THIS SECTION

- [Understanding How to Back Up an Installation on Switches | 310](#)
- [Creating a Snapshot and Using It to Boot a QFX Series Switch | 312](#)
- [Creating a Snapshot and Using It to Boot an EX Series Switch | 317](#)
- [Creating a Snapshot and Using It to Boot an SRX Series Firewall | 318](#)
- [Creating a Snapshot and Using It to Boot an ACX Series Router | 322](#)

The installation process removes all stored files on the device except the `juniper.conf` and SSH files. Therefore, you should back up your current configuration in case you need to return to the current software installation after running the installation program. You can also recover the configuration file and the Junos OS if required.

Understanding How to Back Up an Installation on Switches

IN THIS SECTION

- [Understanding System Snapshot on QFX Switches | 311](#)
- [Understanding System Snapshot on EX Series Switches | 311](#)

You can create copies of the software running on a switch using the system snapshot feature. The system snapshot feature takes a “snapshot” of the files currently used to run the switch—the complete contents of the **/config** and **/var** directories, which include the running Junos OS, the active configuration, and the rescue configuration—and copies all of these files into an alternate (internal, meaning internal flash, or an external, meaning USB flash) memory source. You can then use this snapshot to boot the switch at the next boot up or as a backup boot option.

Understanding System Snapshot on QFX Switches



NOTE: On QFX3500 and QFX3600 switches running Enhanced Layer 2 Software, all of the directories that reside in the “/” partition are read only.



NOTE: System snapshot is not supported on QFX10000 switches.

You can only use snapshots to move files to external memory if the switch was booted from internal memory, or to move files to internal memory if the switch was booted from external memory. You cannot create a snapshot in the memory source that booted the switch even if the snapshot is being created on a different partition in the same memory source.

Snapshots are particularly useful for moving files onto USB flash drives. You cannot use the `copy` command or any other file-moving technique to move files from an internal memory source to USB memory on the switch.

System snapshots on the switch have the following limitations:

- You cannot use snapshots to move files to any destination outside of the switch other than an installed external USB flash drive.
- Snapshot commands are always executed on a local switch.

Understanding System Snapshot on EX Series Switches

The switch can boot from either internal flash media or external (USB) flash media. The contents of the snapshot vary depending on whether you create the snapshot on the media that the switch booted from or on the media that it did not boot from.

Snapshots are particularly useful for moving files onto USB flash drives. You cannot use the `copy` command or any other file-moving technique to move files from an internal memory source to USB memory on the switch.

- If you create the snapshot on the media that the switch did not boot from, the following partitions on the boot media are included in the snapshot: **root**, **altroot**, **var**, **var/tmp**, and **config**.

The **root** partition is the primary boot partition, and the **altroot** partition is the backup boot partition.

- If you create the snapshot on the media that the switch booted from, the root partition that the switch booted from is copied to the alternate root partition. The **var**, **var/tmp**, and **config** partitions are not copied as part of the snapshot because they already exist on the boot media.

The system snapshot feature has the following limitations:

- You cannot use snapshots to move files to any destination outside the switch other than an installed external USB flash drive or switches that are members of the same *Virtual Chassis* as the switch on which you created the snapshot.
- Snapshot commands, like all commands executed on a Virtual Chassis, are executed on the local member switch. If different member switches request the snapshot, the snapshot command is pushed to the Virtual Chassis member creating the snapshot and is executed on that member, and the output is then returned to the switch that initiated the process. For instance, if the command to create an external snapshot on member 3 is entered on member 1, the snapshot of internal memory on member 3 is taken on external memory on member 3. The output of the process is seen on member 1. No files move between the switches.

Creating a Snapshot and Using It to Boot a QFX Series Switch

IN THIS SECTION

- [Creating a Snapshot on an External USB Flash Drive and Using It to Boot a QFX Series Switch | 313](#)
- [Creating a Snapshot and Using It to Boot a QFX3500 and QFX3600 Series Switch | 314](#)

The system snapshot feature takes a “snapshot” of the files currently used to run the device— the complete contents of the **/config** directories, which include the running Juniper Networks Junos OS, the active configuration, and the rescue configuration, as well as the host OS— and copies all of these files into an external USB flash drive.

You can use the snapshot to boot the device at the next bootup or as a backup boot option.

The system snapshot feature is especially effective as a bootup option after a partition corruption, as it is the only recovery option that allows you to completely restore the Junos OS and configuration in the event of a corrupted partition on a switch.



NOTE: EX4600 and most QFX Series switches support snapshot via external USB. (EX4650 switches do not support system snapshot.)



NOTE: The following products do not support system snapshot: QFabric and the QFX5110, QFX5200, and QFX10000 switches.

This topic includes the following tasks:

Creating a Snapshot on an External USB Flash Drive and Using It to Boot a QFX Series Switch

A snapshot can be created on an external USB flash drive after a device is booted using files stored in internal memory.

Ensure that you have the following tools and parts available before creating a snapshot on an external USB flash drive:

- An external USB flash drive that meets the device USB port specifications. See [USB Port Specifications for the QFX Series](#).

To create a snapshot on the external USB flash drive and use it to boot the device:

1. Insert the external USB flash drive.
2. Issue the request system snapshot command.

```
user@device> request system snapshot
fpc0:
-----
Starting snapshot to usb (/dev/da0)
Creating snapshot on the host ..
Copying bootable disk image from host ..
Writing to usb (/dev/da0) ..
Copying 'Host OS' to '/dev/da0s1' .. (this may take a few minutes)
  Copying 'JUNOS' to '/dev/da0s1' .. (this may take a few minutes)
  The following filesystems were archived: / /config Host-OS
```

3. (Optional) Perform this step if you want to boot the device now using the snapshot stored on the external USB flash drive. If you created the snapshot as a backup, do not perform this step.
 - Insert the external USB flash drive.
 - Power cycle the device.

The external USB flash drive is detected.

- The software prompts you with the following options:

```
Junos Snapshot Installer - (c) Juniper Networks 2013
Reboot
Install Junos Snapshot [13.2-20131115_x_132_x51_vjunos.0
Boot to host shell [debug]
```

- Select **Install Junos Snapshot** to install the snapshot located on the external USB flash drive to the device.

The device copies the software from the external USB flash drive, occasionally displaying status messages. When the software is finished being copied from the external USB flash drive to the device, the device then reboots from the internal flash storage on which the software was just installed. When the reboot is complete, the device displays the Junos OS login prompt:

```
root@device#
```

Creating a Snapshot and Using It to Boot a QFX3500 and QFX3600 Series Switch

IN THIS SECTION

- [Creating a Snapshot on a USB Flash Drive and Using It to Boot the Switch | 315](#)
- [Creating a Snapshot on an Internal Flash Drive and Using it to Boot the Switch | 315](#)
- [Creating a Snapshot on the Alternate Slice of the Boot Media | 316](#)

The system snapshot feature takes a “snapshot” of the files currently used to run the QFX Series switch—the complete contents of the **/config** and **/var** directories, which include the running Juniper Networks Junos OS, the active configuration, and the rescue configuration—and copies all of these files into an alternate (internal, meaning internal flash, or an external, meaning USB flash) memory source. You can then use these snapshots to boot the switch at the next bootup or as a backup boot option.

The system snapshot feature is especially effective as a bootup option after a partition corruption, as it is the only recovery option that allows you to completely restore the Junos OS and configuration in the event of a corrupted partition.

This topic includes the following tasks:

Creating a Snapshot on a USB Flash Drive and Using It to Boot the Switch



NOTE: Creating a snapshot is not supported on QFX10000 switches.

A snapshot can be created on USB flash memory after a switch is booted using files stored in internal memory.

Ensure that you have the following tools and parts available before creating a snapshot on a USB Flash drive:

- A USB flash drive that meets the QFX Series switch USB port specifications. See [USB Port Specifications for the QFX Series](#).

To create a snapshot on USB flash memory and use it to boot the switch:

1. Place the snapshot into USB flash memory:

```
user@switch> request system snapshot partition
```



NOTE: This example uses the `partition` option. If you have already created a partition for the snapshot, you don't need to use the `partition` option.

2. (Optional) Perform this step if you want to boot the switch now using the snapshot stored on the external USB flash drive. If you created the snapshot as a backup, do not perform this step.

- To reboot the switch using the most recently created snapshot:

```
user@switch> request system reboot
```

- To reboot the switch using a snapshot in a specific partition on the USB flash drive:

```
user@switch> request system reboot slice 1
```

Creating a Snapshot on an Internal Flash Drive and Using it to Boot the Switch

A snapshot can be created on internal memory after a switch is booted using files stored in external memory.

To create a snapshot in internal memory and use it to boot the switch:

1. Place the snapshot files in internal memory:

```
user@switch> request system snapshot partition
```



NOTE: This example uses the `partition` option. If you have already created a partition for the snapshot, you don't need to use the `partition` option.

2. (Optional) Perform this step if you want to boot the switch now using the newly created snapshot. If you created the snapshot as a backup, do not perform this step.

- To reboot the switch using the most recently created snapshot:

```
user@switch> request system reboot
```

- To reboot the switch using a snapshot in a specific partition in internal memory:

```
user@switch> request system reboot slice 1
```

Creating a Snapshot on the Alternate Slice of the Boot Media

The alternate slice of the boot media contains a backup software image that the switch can boot from if it is unable to boot from the primary slice. When you upgrade software, the new software image gets copied only to the primary slice of the boot media.

To create a snapshot of the currently booted software image on the backup slice of the boot media:

```
user@switch> request system snapshot slice alternate
```

After the system boots up, you will see the following message before the login prompt:

```
WARNING: THIS DEVICE HAS BOOTED FROM THE BACKUP JUNOS IMAGE
```

It is possible that the primary copy of JUNOS failed to boot up properly, and so this device has booted up from the backup copy.

Please re-install JUNOS to recover the primary copy in case it has been corrupted.

The system will generate an alarm indicating that the switch has booted from the backup slice.

Creating a Snapshot and Using It to Boot an EX Series Switch

IN THIS SECTION

- [Creating a Snapshot on a USB Flash Drive and Using It to Boot the Switch](#) | 317

The system snapshot feature takes a “snapshot” of the files currently used to run the switch and copies them to an alternate storage location. You can then use this snapshot to boot the switch at the next bootup or as a backup boot option.

This topic includes the following tasks:

Creating a Snapshot on a USB Flash Drive and Using It to Boot the Switch

You can create a snapshot on USB flash memory after a switch is booted by using files stored in internal memory.

Ensure that you have the following tools and parts available before creating a snapshot on a USB flash drive:

- A USB flash drive that meets the switch USB port specifications. See [USB Port Specifications for an EX Series Switch](#).

To create a snapshot on USB flash memory and use it to boot the switch:

1. Place the snapshot into USB flash memory:

```
user@switch> request system snapshot partition media usb
```

2. (Optional) Perform this step if you want to boot the switch now using the snapshot stored on the USB flash drive.

```
user@switch> request system reboot media usb
```

Creating a Snapshot and Using It to Boot an SRX Series Firewall

IN THIS SECTION

- [Create a Snapshot and Using It to Boot an SRX Series Firewall | 318](#)
- [Backup the Current Installation on SRX Series Firewalls | 320](#)

Create a Snapshot and Using It to Boot an SRX Series Firewall

IN THIS SECTION

- [Requirements | 318](#)
- [Overview | 318](#)
- [Configuration | 319](#)

This example shows how to configure a boot device.

Requirements

Before you begin, ensure that the backup device has a storage capacity of at least 1 GB. See ["Ensure Sufficient Disk Space for Junos OS Upgrades on SRX Series Firewalls" on page 64](#).

Overview

You can configure a boot device to replace the primary boot device on your SRX Series Firewall or to act as a backup boot device. Use either the J-Web user interface or the CLI to take a snapshot of the configuration currently running on the device, or of the original factory configuration and a rescue configuration, and save it to an alternate medium.

For media redundancy, we recommend that you keep a secondary storage medium attached to the SRX Series Firewall and updated at all times.

If the primary storage medium becomes corrupted and no backup medium is in place, you can recover the primary internal media from the TFTP installation.

You can also configure a boot device to store snapshots of software failures for use in troubleshooting.

You cannot copy software to the active boot device.

After a boot device is created with the default factory configuration, it can operate only in an internal media slot.

This example configures a boot device to back up the currently running and active file system partitions by rebooting from internal media and including only files shipped from the factory.

Configuration

IN THIS SECTION

- [Procedure | 319](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

From operational mode, enter:

```
user@host> request system snapshot partition media internal factory
```

GUI Quick Configuration

Step-by-Step Procedure

To configure a boot device:

1. In the J-Web user interface, select **Maintain>Snapshot**.
2. On the Snapshot page, specify the boot device to copy the snapshot to. From the Target Media list, select the **internal** boot device.
3. Select the Factory check box to copy only default files that were loaded on the internal media when it was shipped from the factory, plus the rescue configuration if one has been set.
4. Select the Partition check box to partition the medium that you are copying the snapshot to. This process is usually necessary for boot devices that do not already have software installed on them.

5. Click **Snapshot**.
6. Click **OK** to check your configuration and save it as a candidate configuration.
7. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

1. To configure a boot device:

```
user@host> request system snapshot partition media internal factory
```

Results

From configuration mode, confirm your configuration by entering the `show system snapshot media internal` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host> show system snapshot media internal
```

```
Information for snapshot on      internal (/dev/ad0s1a) (backup)
Creation date: Oct 9 13:30:06 2009
JUNOS version on snapshot:
  junos   : 10.0B3.10-domestic
Information for snapshot on      internal (/dev/ad0s2a) (primary)
Creation date: Jan 6 15:45:35 2010
JUNOS version on snapshot:
  junos   : 10.2-20091229.2-domestic
```

If you are done configuring the device, enter `commit` from configuration mode.

Backup the Current Installation on SRX Series Firewalls

IN THIS SECTION

-  [Backup the Current Installation on SRX5800, SRX5600, and SRX5400 Devices | 321](#)

- [Backup the Current Installation on SRX300, SRX320, SRX340, SRX345, SRX380, SRX3400, and SRX3600 Devices | 321](#)

This topic includes the following sections:

Backup the Current Installation on SRX5800, SRX5600, and SRX5400 Devices

Back up the current installation so that you can return to the current software installation. The installation process using the installation package (jinstall*, for example) removes all stored files on the device except the juniper.conf and SSH files. Therefore, you should back up your current configuration in case you need to return to the current software installation after running the installation program.

To back up Junos OS on the SRX Series Firewalls, issue the `request system snapshot` CLI operational mode command. This command saves the current software installation on the hard disk, external USB storage media device, or solid-state drive (SSD).

When the `request system snapshot` command is issued, the `/root` file system is backed up to `/altroot`, and `/config` is backed up to `/altconfig`. The `/root` and `/config` file systems are on the devices's CompactFlash card, and the `/altroot` and `/altconfig` file systems are on the devices's hard disk or solid-state drive (SSD). When the backup is completed, the current and backup software installations are identical.

To copy the files to the device's hard disk or solid-state drive (SSD), use the following command:

```
user@host> request system snapshot media
```

Backup the Current Installation on SRX300, SRX320, SRX340, SRX345, SRX380, SRX3400, and SRX3600 Devices

On SRX Series Firewalls, you can backup the current Junos OS image and configuration files onto a media (such as a USB or CompactFlash) so that you can retrieve it back if something goes wrong.

To back up the currently running and active file system partitions on the device, use the following command:

```
user@host> request system snapshot media
```

The following options are supported:

- `internal`— Copies the snapshot to internal media.
- `usb`— Copies the snapshot to the USB storage device. This is the default option for SRX300, SRX320, SRX340, SRX345, and SRX380 devices.

Creating a Snapshot and Using It to Boot an ACX Series Router

IN THIS SECTION

- [Understanding System Snapshot on an ACX Series Router | 322](#)
- [Example: Taking a Snapshot of the Software and Configuration | 323](#)

Understanding System Snapshot on an ACX Series Router

The system snapshot feature enables you to create copies of the software running on an ACX Series router. You can use the system snapshot feature to take a “snapshot” of the files currently used to run the router—the complete contents of the root (/) and /config directories, which include the running Juniper Networks Juniper operating system (Junos OS) and the active configuration—and copy all of these files to another media, such as a universal serial bus (USB) storage device, the active slice of a dual-root partitioned router, or the alternate slice of a dual-root partitioned router.



NOTE: Junos OS automatically uses the backup software if the currently running software goes bad. For example, if the da0s1 slice goes bad, Junos OS automatically comes up using the da0s2 slice, and takes a snapshot of the da0s2 slice and copies it to the da0s1 slice if the auto snapshot functionality is configured, which is disabled by default. However, you can also do this manually using the system snapshot feature.



NOTE: In ACX5048 and ACX5096 routers, the system snapshot feature is applicable only when a USB storage device is used.

Typically, you can take a snapshot prior to the upgrade of an image on the dual internal NAND flash device (da0s1 or da0s2), or to remedy a bad image, thereby preventing the bad image from rendering the system useless. A snapshot to another media ensures that the device can boot from the other media in case the system does not boot up from the current image.

You can take a snapshot of the currently running software and configuration on a router in the following situations:

- The router's active slice (for example, da0s1) is updated with a new Junos OS image (using the jinstall package). In such a case, you must update the other slice (da0s2) with the new image.



NOTE: The active slice can be da0s1 or da0s2.

- The router's active slice (for example, da0s1) is corrupted and the router is rebooted from the backup slice (that is, from da0s2). Therefore, you must restore a new image on the active slice—that is, on da0s1.
- Both slices of the router's dual internal NAND flash device are corrupted and the router continues trying to reboot. In this situation, you can insert a USB storage device, boot the router from that device, and restore the NAND flash device slices—da0s1 and da0s2.



NOTE: Before you attempt to take a snapshot from the USB storage device, ensure that the USB storage device contains an image of Junos OS from which the router can boot up.

SEE ALSO

[request system snapshot \(ACX Series\)](#)

Example: Taking a Snapshot of the Software and Configuration

IN THIS SECTION

- [Taking a Snapshot](#) | 323

This example includes six scenarios in which you can take a snapshot of the currently running software and configuration on an ACX Series router, prior to the upgrade of an image or to remedy a bad image, thereby preventing the bad image from rendering the system useless.

Taking a Snapshot

Step-by-Step Procedure

Scenario: To take a snapshot from a NAND flash device slice to a USB storage device:

1. Boot up the router from the NAND flash device and make sure that a formatted USB storage device is plugged in to the router's USB port. The USB storage device must be formatted for the root (/) and /config directories.

2. Issue the request system snapshot command.

```

user@host> request system snapshot
Verifying compatibility of destination media partitions...
Running newfs (254MB) on usb media / partition (da1s1a)...
Running newfs (47MB) on usb media /config partition (da1s1e)...
Copying '/dev/da0s2a' to '/dev/da1s1a' .. (this may take a few minutes)
Copying '/dev/da0s2e' to '/dev/da1s1e' .. (this may take a few minutes)
The following filesystems were archived: / /config

```

The root (/) and /config directories from the currently mounted NAND flash slice are copied to the USB storage device.

Step-by-Step Procedure

Scenario: To take a snapshot from a NAND flash device slice to a USB storage device with formatting:

1. Boot up the router from the NAND flash device and make sure that a USB storage device is plugged in to the router's USB port.



NOTE: Formatting a USB storage device deletes all the data on the USB storage device.

2. Issue the request system snapshot partition command.

```

user@host> request system snapshot partition
clearing current label...
Partitioning usb media (da1) ...
Partitions on snapshot:

      Partition Mountpoint Size  Snapshot argument
      -
      a         /           312MB root-size
      e         /config    47MB  config-size
      f         /var       620MB var-size

Running newfs (312MB) on usb media / partition (da1s1a)...
Running newfs (47MB) on usb media /config partition (da1s1e)...
Running newfs (620MB) on usb media /var partition (da1s1f)...
Copying '/dev/da0s2a' to '/dev/da1s1a' .. (this may take a few minutes)
Copying '/dev/da0s2e' to '/dev/da1s1e' .. (this may take a few minutes)
The following filesystems were archived: / /config

```

After the USB storage device is formatted, the root (/) and /config directories from the currently mounted NAND flash slice are copied to the USB storage device.

Step-by-Step Procedure

Scenario: To take a snapshot from the active slice of the NAND flash device to the alternate slice:

1. Boot up the router from the NAND flash device.
2. Issue the request system snapshot slice alternate command.

```
user@host> request system snapshot slice alternate
Verifying compatibility of destination media partitions...
Running newfs (439MB) on internal media / partition (da0s1a)...
Running newfs (46MB) on internal media /config partition (da0s1e)...
Copying '/dev/da0s2a' to '/dev/da0s1a' .. (this may take a few minutes)
Copying '/dev/da0s2e' to '/dev/da0s1e' .. (this may take a few minutes)
The following filesystems were archived: / /config
```

The root (/) and /config directories from the currently mounted NAND flash slice are copied to the other slice.

Step-by-Step Procedure

Scenario: To take a snapshot from an active slice of the NAND flash device to the alternate slice after partitioning:

1. Boot up the router from the NAND flash device.
2. Issue the request system snapshot partition slice alternate command.

```
user@host> request system snapshot partition slice alternate
Verifying compatibility of destination media partitions...
Running newfs (439MB) on internal media / partition (da0s1a)...
Running newfs (46MB) on internal media /config partition (da0s1e)...
Copying '/dev/da0s2a' to '/dev/da0s1a' .. (this may take a few minutes)
Copying '/dev/da0s2e' to '/dev/da0s1e' .. (this may take a few minutes)
The following filesystems were archived: / /config
```

The BSD label (disk partitioning information) for the active flash slice is installed and then the root (/) and /config directories from the currently mounted NAND flash slice are copied to the other slice.

Step-by-Step Procedure

Scenario: To take a snapshot from a USB storage device to the active slice of the NAND flash device:

1. Boot up the router from a USB storage device containing the required Junos OS image.
2. Issue the `request system snapshot` command.

```
user@host> request system snapshot
Verifying compatibility of destination media partitions...
Running newfs (439MB) on internal media / partition (da0s1a)...
Running newfs (46MB) on internal media /config partition (da0s1e)...
Copying '/dev/da1s1a' to '/dev/da0s1a' .. (this may take a few minutes)
Copying '/dev/da1s1e' to '/dev/da0s1e' .. (this may take a few minutes)
The following filesystems were archived: / /config
```

The root (/) and /config directories from the USB storage device are copied to the active NAND flash slice.

Step-by-Step Procedure

Scenario: To take a snapshot from a USB storage device to the active slice of the NAND flash device after partitioning:

1. Boot up the router from a USB storage device containing the required Junos OS image.
2. Issue the `request system snapshot partition` command.

```
user@host> request system snapshot partition
Verifying compatibility of destination media partitions...
Running newfs (439MB) on internal media / partition (da0s1a)...
Running newfs (46MB) on internal media /config partition (da0s1e)...
Copying '/dev/da1s1a' to '/dev/da0s1a' .. (this may take a few minutes)
Copying '/dev/da1s1e' to '/dev/da0s1e' .. (this may take a few minutes)
The following filesystems were archived: / /config
```

The BSD label (disk partitioning information) for the active flash slice is installed and then the root (/) and /config directories from the USB storage device are copied to the active NAND flash slice.

SEE ALSO

[Understanding System Snapshot on an ACX Series Router](#)

[request system snapshot \(ACX Series\)](#)

Recover Junos OS

IN THIS CHAPTER

- [Recovery of Junos OS | 328](#)
- [Recovery Using an Emergency Boot Device \(Junos OS\) | 334](#)
- [How to Recover Junos OS with Upgraded FreeBSD | 343](#)
- [Rescue and Recovery of Configuration File \(Junos OS\) | 351](#)
- [Autorecovery of Configuration, Licenses, and Disk Information on SRX Series Devices | 360](#)

Recovery of Junos OS

SUMMARY

In case of a failed software installation or a failure after installing Junos OS, such as the CLI not working, you can recover the failed software. You can recover the software by installing Junos OS and remove the existing Junos OS image to install a new image.

IN THIS SECTION

- [Recovering from a Failed Software Installation | 328](#)
- [Recovering Junos OS on a Device Running Junos OS with Upgraded FreeBSD | 331](#)

Recovering from a Failed Software Installation

IN THIS SECTION

- [Problem | 329](#)
- [Solution | 329](#)

Problem

Description

If the Junos OS appears to have been installed but the CLI does not work, or if the device has no software installed, you can use this recovery installation procedure to install the Junos OS.

Solution

If a Junos OS image already exists on the switch, you can either install the new Junos OS package in a separate partition, in which case both Junos OS images remain on the switch, or you can remove the existing Junos OS image before you start the new installation process.



NOTE: QFX5100, QFX5200, EX4600, and QFX10000 switches do not have a separate partition to reinstall a Junos OS image.

A recovery image is created automatically on these switches. If a previously-running switch is powered on and unable to boot using a Junos OS image, you can boot the switch using the recovery Junos OS image by selecting an option in the “Select a recovery image” menu.

We suggest creating a system snapshot on your switch onto the external USB flash drive, and using the snapshot for recovery purposes. The system snapshot feature takes a “snapshot” of the files currently used to run the device—the complete contents of the **/config** directories, which include the running Juniper Networks Junos OS, the active configuration, and the rescue configuration, as well as the host OS—and copies all of these files into an external USB flash drive. See [Creating a Snapshot and Using It to Boot a QFX3500 and QFX3600 Series Switch](#) or [Creating a Snapshot and Using It to Boot a QFX Series Switch](#).

System snapshot is not supported on QFX5200 and QFX10000 switches.

To perform a recovery installation:

1. Power on the switch. The loader script starts.
2. After the message **Loading /boot/defaults/loader.conf** appears, you are prompted with the following message:

Hit [Enter] to boot immediately, or space bar for command prompt.

Press the Spacebar to enter the manual loader. The `loader>` prompt appears.



NOTE: The loader prompt does not appear on QFX5100, QFX5200, EX4600, and QFX10000 switches.

On QFX5100, QFX5200, EX4600, and QFX10000 switches only, a recovery image is automatically saved if a previously-running switch is powered on and unable to boot using a Junos OS image.

The “Select a recovery image” menu appears on the console when one of these switches is booted and unable to load a version of Junos OS. Follow the instructions in the “Select a recovery image” menu to load the recovery version of Junos OS for one of these switches.

You can ignore the remainder of this procedure if you are using a QFX5100, QFX5200, EX4600, or QFX10000 switch.

3. Enter the following command:

```
loader> install [--format] [--external] source
```

where:

- **format**—Enables you to erase the installation media before installing the installation package. If you do not include this option, the system installs the new Junos OS in a different partition from that of the most recently installed Junos OS.
- **external**—Installs the installation package onto external media (a USB stick, for example).
- **source**—Represents the name and location of the Junos OS package, either on a server on the network or as a file on an external media, as shown in the following two examples:
 - Network address of the server and the path on the server; for example, **tftp://192.0.2.0/junos/jinstall-qfx-5e-flex-15.1X53-D30.5-domestic-signed.tgz**
 - Junos OS package on a USB device (commonly stored in the root drive as the only file), for example, **file://jinstall-qfx-5e-flex-15.1X53-D30.5-domestic-signed.tgz**.

The installation now proceeds normally and ends with a login prompt.

SEE ALSO

[Creating a Snapshot and Using It to Boot a QFX3500 and QFX3600 Series Switch](#)

[Creating a Snapshot and Using It to Boot a QFX Series Switch](#)

Recovering Junos OS on a Device Running Junos OS with Upgraded FreeBSD

Certain hardware platforms run an upgraded FreeBSD kernel (FreeBSD 10.x or later) instead of FreeBSD 6.1. Juniper Networks devices that run Junos OS with upgraded FreeBSD have two separate volumes:

- **dev/gpt/junos** (**/junos** for short) volume that is used to run Junos OS and to store the configuration and log files
- **dev/gpt/oam** (**/oam** for short), an Operations, Administration, and Maintenance (OAM) volume that is used to store a complete backup of Junos OS and the configuration.

In case of damage to the device's software or failure of the **/junos** volume, you can use the backed up software and configuration stored in the **/oam** volume to boot the system and restore Junos OS with the recovery configuration. To perform this reboot and restore the configuration, the **/oam** volume must have all of the information required to provide the system with a running configuration. This information is provided by the recovery snapshot, created using the `request system snapshot recovery` command.



NOTE: You need console access to perform the following procedure to recover Junos OS.

To recover Junos OS by using the recovery snapshot stored in the **/oam** volume:

1. Power off the device, such as a router or a switch, by pressing the power button on the front panel.
2. Connect and configure the management device, such as a PC or a laptop, as follows:
 - a. Turn off the power to the management device.
 - b. Plug one end of the Ethernet rollover cable supplied with the device into the RJ-45-to-DB-9 serial port adapter supplied with the device.
 - c. Plug the RJ-45-to-DB-9 serial port adapter into the serial port on the management device.
 - d. Connect the other end of the Ethernet rollover cable to the console port on the device.
 - e. Turn on the power to the management device.
 - f. On the management device, start your asynchronous terminal emulation application (such as Microsoft Windows Hyperterminal) and select the appropriate communication (COM) port to use (for example, COM1).
 - g. Configure the port settings as follows:
 - Bits per second: 9600
 - Data bits: 8
 - Parity: None

- Stop bits: 1
- Flow control: None

3. Power on the device by pressing the power button on the front panel.

Verify that the **POWER** LED on the front panel turns green.

The terminal emulation screen on your management device displays the boot sequence of the device.

4. Access the Junos Main Menu.

Press Ctrl+c within the 3-second window to stop the automatic boot sequence and display the Junos Main Menu.

```
Main Menu

1. Boot [D]irectly fromJunos volume
2. Boot Junos volume in [S]afe mode

3. [R]eboot

4. [B]oot menu
5. [M]ore options

Choice:
```

5. At the Choice: prompt in Junos Main Menu, enter **B** or **4** to choose 4. [B]oot menu. Only the options valid for your device appear. The menu numbers change based on what's available.

```
Boot Menu

1. Boot [P]revious installed Junos packages
2. Boot [A]ctive Junos packages
3. Boot Junos in [S]ingle user mode
4. Boot from [R]ecover snapshot
5. Boot from [Network]
6. Boot from [U]SB

7. Boot to [O]AM shell

8. Snapshot [B]oot menu

9. [M]ain menu
```

Choice:

6. At the Choice: prompt in Boot Menu, enter **R** or **3** to choose the 4. Boot from [R]ecovery snapshot option. The device reboots into recovery mode. The following sample output shows the messages displayed on the terminal when you recover Junos OS on an EX2300 switch.

```

Booting from recovery snapshot ...
-
/boot/junos/boot/os-kernel/kernel data=0xe8c000 syms=[0x4+0x6b020+0x4+0x72cfe]
/boot/junos/boot/os-kernel/ex2300-48mp.dtb size=0x18b8
/boot/junos/boot/os-kernel/ex2300.dtb size=0x1e67
/boot/junos/boot/junos-modules/fips_core.ko text=0x13bc data=0x275+0x7
syms=[0x4+0x7a0+0x4+0x518]
loading required module 'netstack'
/boot/junos/boot/netstack/netstack.ko text=0x910a3c data=0x3ae2f+0x10dded
syms=[0x4+0xf0570+0x4+0xdc394]
loading required module 'crypto'

[...Output truncated...]

/var/pdb/profile_db
initialized

Profile database initialized
realpath: /dev/dumpdev: No such file or directory
/etc/rc: WARNING: Dump device does not exist. Savecore not run.
Prefetching /usr/sbin/rpd ...
Prefetching /usr/sbin/lacpd ...
Prefetching /usr/sbin/chassisd ...
mkdir: /packages/sets/active: Read-only file system
Starting jlaunchhelperd.
sysctl: unknown oid 'kern.rtc_retries'
Starting
cron.

Fri Jun 22 01:25:20 PDT
2018

FreeBSD/arm (device-name)

```

```
(ttyu0)
login:
```

7. Log in to the device and run the command `request system recover junos-volume`.

```
[...Output truncated...]
login: root

--- JUNOS 18.1-20180125.0 built 2018-01-25 20:34:55 UTC

root@RE:0% FreeBSD/arm (Amnesiac) (ttyu0)

login: root

--- JUNOS
Note: Junos is currently running in recovery mode on the OAM volume
root@RE:0% CLI
{master:0}
root> request system recover junos-volume
NOTICE: Recovering the Junos volume ...
...
```

8. Reboot the device to finish the recovery.

```
root> request system reboot junos
```



NOTE: For more information see [KB32642](#).

Recovery Using an Emergency Boot Device (Junos OS)

SUMMARY

If Junos OS software is damaged on your device, the emergency boot device helps you to recover the software.

IN THIS SECTION

- [Creating an Emergency Boot Device for Routers | 335](#)

- [Creating an Emergency Boot Device for QFX Series Switches | 337](#)
- [Recovering the Installation Using an Emergency Boot Device on QFX Series Switches | 338](#)
- [Performing a Recovery Installation | 341](#)

Creating an Emergency Boot Device for Routers

If the device's Junos OS software is damaged in some way that prevents Junos OS software from loading completely, you can use the emergency boot device to revive the device. The emergency boot device repartitions the primary disk and reloads a fresh installation of Junos OS software.

Certain hardware platforms run an upgraded FreeBSD kernel (FreeBSD 10.x or later) instead of FreeBSD 6.1. To determine if your system uses Junos OS with upgraded FreeBSD, see: [Junos kernel upgrade to FreeBSD 10+](#).

The procedures outlined in this section discuss how to create an emergency boot device for any ACX Series or MX Series router.



NOTE: For devices with Routing Engines with VMhost support, you can use the USB device as an emergency boot device. For more information, see "[Creating an Emergency Boot Device for Routing Engines with VM Host Support](#)" on page 416. After the `junos-vmhost-install-usb` image is written to the USB drive using the `dd` command, you can boot to the USB using the `request vmhost reboot usb` command.

To create an emergency boot device:

1. Use FTP to copy the installation media into the router's `/var/tmp` directory.
2. Insert the PC Card into the external PC Card slot or USB storage device into the USB port.
3. In the UNIX shell, navigate to the `/var/tmp` directory:

```
start shell
cd /var/tmp
```

4. Log in as su:

```
su [enter]
password: [enter SU password]
```

5. For Junos OS with upgraded FreeBSD only, expand the image, for example:

```
gzip -d installMedia.img.gz
```

where *installMedia* refers to the installation media downloaded into the **/var/tmp** directory. For example, for Junos OS with upgraded FreeBSD, the filename might be **junos-install-media-usb-mx-x86-64-16.1R2.11.img.gz**. (To determine if your system uses Junos OS with upgraded FreeBSD, see: [Junos kernel upgrade to FreeBSD 10+](#).)

6. Issue the following commands:

- For Junos OS with upgraded FreeBSD:

```
dd if=/dev/zero of=/dev/externalDrive count=20
dd if=installMedia.img of=/dev/externalDrive bs=256k
```

- For Junos OS:

```
dd if=/dev/zero of=/dev/externalDrive count=20
dd if=installMedia of=/dev/externalDrive bs=64k
```

where:

- *externalDrive*—Refers to the removable media name of the emergency boot device. For the names of the removable media, see the table in "[Routing Engines and Storage Media Names \(ACX Series, M Series, MX Series, PTX Series, T Series, TX Matrix, TX Matrix Plus, and JCS 1200 Routers\)](#)" on [page 32](#).
- *installMedia*—Refers to the installation media downloaded into the **/var/tmp** directory. For example, the filename might be **install-media-9.0R2.10-domestic** for Junos OS or, for Junos OS with upgraded FreeBSD, **junos-install-media-usb-mx-x86-64-16.1R2.11.img** (unzipped). (To determine if your system uses Junos OS with upgraded FreeBSD, see: [Junos kernel upgrade to FreeBSD 10+](#).)

7. Log out as su:

```
exit
```

Creating an Emergency Boot Device for QFX Series Switches

Before you begin, you need to download the installation media image for your device and Junos OS release from <https://www.juniper.net/customers/support/>.

If Junos OS on the device is damaged in some way that prevents the software from loading properly, you can use an emergency boot device to repartition the primary disk and load a fresh installation of Junos OS. Use the following procedure to create an emergency boot device.



NOTE: You can create the emergency boot device on another Juniper Networks device, or any laptop or desktop PC that supports Linux. The steps you take to create the emergency boot device vary, depending on the device.

To create an emergency boot device:

1. Use FTP to copy the installation media image into the **/var/tmp** directory on the device.
2. Insert a USB storage device into the USB port.
3. From the CLI, start the shell:

```
user@device> start shell
%
```

4. Use the gunzip command to unzip the image file.
5. Switch to the root account using the su command:

```
% su
Password: password
```



NOTE: The password is the root password for the device. If you logged in to the device as the root user, you do not need to perform this step.

6. Enter the following command on the device:

```
root@device% dd if=/var/tmp/filename of=/dev/da0 bs=1m
```

The device writes the installation media image to the USB storage device:

```
root@device% dd if=install-media-qfx-5e-15.1X53-D30.5-domestic.img of=/dev/da0 bs=1m
1399+0 records in
1399+0 records out
1466957824 bytes transferred in 394.081902 secs (3722469 bytes/sec)
```

7. Log out of the shell:

```
root@device% exit
% exit
user@device>
```

Recovering the Installation Using an Emergency Boot Device on QFX Series Switches

If Junos OS on your device is damaged in some way that prevents the software from loading correctly, you may need to perform a recovery installation using an emergency boot device (for example, a USB flash drive) to restore the default factory installation. Once you have recovered the software, you need to restore the device configuration. You can either create a new configuration as you did when the device was shipped from the factory, or if you saved the previous configuration, you can simply restore that file to the device.

If at all possible, you should try to perform the following steps before you perform the recovery installation:

1. Ensure that you have an emergency boot device to use during the installation. See ["Creating an Emergency Boot Device for QFX Series Switches" on page 337](#) for information on how to create an emergency boot device.
2. Copy the existing configuration in the file `/config/juniper.conf.gz` from the device to a remote system, such as a server, or to an emergency boot device. For extra safety, you can also copy the backup configurations (the files named `/config/juniper.conf.n`, where *n* is a number from 0 through 9) to a remote system or to an emergency boot device.

You can use the system snapshot feature to complete this step. The system snapshot feature takes a "snapshot" of the files currently used to run the QFX Series switch—the complete contents of the `/config` and `/var` directories, which include the running Junos OS, the active configuration, and the rescue configuration—and copies all of these files into a memory source. See ["Creating a Snapshot and Using It to Boot a QFX Series Switch" on page 312](#).



NOTE: System snapshot is not supported on QFX10000 and QFX5200 switches.



CAUTION: The recovery installation process completely overwrites the entire contents of the internal flash storage.

3. Copy any other stored files to a remote system as desired.

To reinstall Junos OS:

1. Connect to the device's console port (either directly or through a console server).
2. Insert the emergency boot device into the QFX Series switch.
3. Reboot or power cycle the device.
4. As soon as the device reboots, keep pressing **Esc** until the boot options menu opens.

```
Press Esc for boot options
...
Intel(R) Xeon(R) CPU @ 2.50GHz          2.50 GHz
V0018.8                               16384 MB RAM
Continue
Boot Manager
Device Manager
Boot From File
Setup Utility
```



NOTE: You might have to reboot or power cycle the device more than once if you miss hitting **Esc** to open the boot options menu.

5. In the boot options menu, select **Boot Manager**.
6. In the Boot Manager menu, select the emergency boot device. In this example, the emergency boot device is the USB device.



NOTE: In later releases, the Boot Manager menu might display two different entries for the same USB recovery device. Select the **EFI USB Device** entry.

Boot Manager

Boot Option Menu

```

SSD0 : ATP M.2 2242
IBA GE Slot 0101 v1350
IBA GE Slot 0102 v1350
USB : General Udisk
SSD1 : ATP M.2 2242
IBA GE Slot 0103 v1350
EFI HDD Device (ATP M.2 2242)
Internal EFI Shell

```

The Juniper Linux Installer or GNU GRUB menu opens. The menu and options may differ slightly depending on the platform and release.

7. If you have Junos OS software from the factory installed on the emergency boot device, the software prompts you with the following options:

```

Juniper Linux Installer - (c) Juniper Networks 2014
    Reboot
    Install Juniper Linux Platform
    Boot to host shell [debug]

```

Select **Install Juniper Linux Platform** to install the Junos OS software from the emergency boot device.



NOTE: Depending on the platform and release, you may see different entries such as **Install Juniper Linux**, **Install Juniper Linux Platform**, or **Install Juniper Linux with secure boot support**.

8. The device copies the software from the emergency boot device, occasionally displaying status messages. Copying the software can take up to 12 minutes.
9. After the software is copied to the device, the device reboots from the internal flash storage on which the software was just installed.



NOTE: If the Boot Manager menu includes both SSD drive and EFI HDD Device entries, manually select the **EFI HDD Device** option.

When the reboot is complete, the device displays the Junos OS login prompt:

```
root@switch#
```

10. Create a new configuration as you did when the device was shipped from the factory, or restore the previously saved configuration file to the device.
11. Remove the emergency boot device.

Performing a Recovery Installation

If the device's software is corrupted or otherwise damaged, you may need to perform a recovery installation, using the emergency boot device to restore the default factory installation. Once you have recovered the software, you will need to restore the router or switch's configuration. You can either create a new configuration as you did when the device was shipped from the factory, or if you saved the device's previous configuration, you can simply restore that file to the system.

Depending on the situation, you should try to perform the following steps before you perform the recovery installation:

1. Ensure you have an emergency recovery disk to use during the installation. When the device is first shipped, an emergency recovery disk is provided with it. For instructions on creating an emergency boot device, see ["Creating an Emergency Boot Device for Routers" on page 335](#)
2. Copy the existing configuration in the file `/config/juniper.conf.gz` from the device to a remote system. For extra safety, you can also copy the backup configurations (the files named `/config/juniper.conf.n`, where *n* is a number from 0 through 9).



CAUTION: The recovery installation process completely overwrites the entire contents of the fixed storage media.

3. Copy any other stored files to a remote system as desired.

To reinstall Junos OS:

1. Insert the removable media emergency boot device into the device.



NOTE: You can store a configuration on installation media such as a PC Card or USB stick.

2. Reboot the device.

If the CLI is still active, issue the **request system reboot** command from command mode to reboot the device.

If the CLI is not working, manually power off the device using the main power switch, wait 10 seconds, and then power the device back on.

3. When the software prompts you with the following question, type **y**:



NOTE: Junos OS with upgraded FreeBSD does not display the following warning. To determine if your system uses Junos OS with upgraded FreeBSD, see: [Junos kernel upgrade to FreeBSD 10+](#).

```
WARNING: The installation will erase the contents of your disk. Do you wish to continue (y/n)? y
```

The device copies the software from the removable media emergency boot device onto your system, occasionally displaying status messages. Copying the software can take up to 45 minutes, depending on the device. When the process is complete, the system boots into Amnesiac state and the login prompt is displayed.

4. Remove the removable media emergency boot device.



NOTE: For SRX300, SRX320, SRX340, SRX345, and SRX380 firewalls: When one of these firewalls is already booted with a recovery image, you cannot recover the firewall using a `junos-install-media-usb*` image from an already-inserted bootable USB device. If you unplug the bootable USB device and plug the device back in, then you can install the recovery image from the USB device. This behavior is designed to avoid the scenario where someone installs a recovery image and issues a reboot command for some other purpose, but the device copies and installs the second image from the USB device instead of using the original recovery image.

5. Log in as root on the device's console port and issue the **request system reboot** command from command mode to reboot the device.

The device reboots from the boot device on which the software was just installed. When the reboot is complete, the device displays the login prompt.

6. Create a new configuration as you did when the device was shipped from the factory, or restore a previously saved configuration file to the system. For more information, see *Configuring Junos OS for the First Time on a Device with a Single Routing Engine*, *Configuring Junos OS for the First Time on a Device with Dual Routing Engines*, and ["Restoring a Saved Configuration" on page 357](#).

How to Recover Junos OS with Upgraded FreeBSD

SUMMARY

When you don't have access to the CLI, you can use the Junos Main Menu and Boot Menu to recover a device running Junos OS with Upgraded FreeBSD.

IN THIS SECTION

- [Ways to Recover Junos OS with Upgraded FreeBSD Without the Use of the CLI | 343](#)
- [How to Access the Junos Main Menu, Boot Menu, and Options Menu | 347](#)

Ways to Recover Junos OS with Upgraded FreeBSD Without the Use of the CLI

IN THIS SECTION

- [Boot from the /junos Volume | 344](#)
- [Boot from Safe Mode | 344](#)
- [Boot from a Previously Installed Release of Junos OS with Upgraded FreeBSD | 344](#)
- [Boot into Single-User Mode | 345](#)
- [Boot from a Recovery Snapshot | 345](#)
- [Boot from a USB Device | 346](#)
- [Boot to the OAM Shell | 346](#)
- [CLI Recovery Mode | 346](#)
- [Check File System | 346](#)
- [Enable/Disable Verbose Boot | 346](#)
- [Boot Prompt | 347](#)

If a device running Junos OS with upgraded FreeBSD has a damaged operating system or configuration that prevents the system from booting normally, or you need to recover the root password, the CLI is unavailable to you. But you can access and use the Junos Main Menu and Boot Menu. These menus have options such as booting from a USB device or a previously installed version of Junos OS, or using CLI Recovery mode to change your root password. To determine if your system uses Junos OS with upgraded FreeBSD, see: [Junos kernel upgrade to FreeBSD 10+](#).



NOTE: For SRX300, SRX320, SRX340, SRX345, and SRX380 firewalls upgraded to Junos OS Release 24.4R1, see *KB85650* for information on how to recover a system that is not booting normally. For these firewalls, you cannot use the Boot menu to boot from a USB device.

Boot from the /junos Volume

Juniper Networks devices that run Junos OS with upgraded FreeBSD have two separate volumes:

- **dev/gpt/junos (/junos for short)** volume that is used to run Junos OS and to store the configuration and log files
- **dev/gpt/oam (/oam for short)**, an Operations, Administration, and Maintenance (OAM) volume that is used to store a complete backup of Junos OS and the configuration.

If a device running Junos OS with upgraded FreeBSD has a damaged operating system or configuration, preventing the system from booting normally, you can still boot from the **/junos** volume without using the CLI command `request system reboot`. Access the Junos Main Menu. Booting the **/junos** volume is the first option on the Junos Main Menu.

Boot from Safe Mode

Safe mode is a diagnostic mode of a computer's operating system that has reduced functionality, making the task of isolating problems easier since many non-core components are disabled. In Junos OS with upgraded FreeBSD, safe mode boots the entire Junos OS and FreeBSD but with a few kernel features disabled.

One other difference between normal mode and safe mode is that for EX3400 devices, symmetric multiprocessing (SMP) in normal mode uses a dual core, whereas in safe mode, it uses a single core.

An installation that has a major problem (such as disk corruption or the installation of poorly configured software) that prevents the operating system from booting into its normal operating mode may boot in safe mode and allow you to diagnose the problem.

Booting from Safe Mode is the second option on the Junos Main Menu.

Boot from a Previously Installed Release of Junos OS with Upgraded FreeBSD

With devices running Junos OS with upgraded FreeBSD, you can boot from a previous release of the OS, provided there was a previous image on the device and it is still there. Previous image files can be found in the **/packages/sets/previous** directory. Some platforms do not keep an older image due to storage space limitations (for example, EX2300 and EX3400 do not have a **/packages/sets/previous** directory).

The following is sample output from an EX9200 switch, showing the previous image:

```
root@:/ # ls -al /packages/sets/previous/
total 20
drwxr-xr-x  4 root  wheel  1536 Mar 30 15:45 .
drwxrwxrwx  4 root  wheel   512 Mar 30 18:47 ..
drwxr-xr-x  2 root  wheel   512 Mar 30 15:45 boot
lrwxr-xr-x  1 root  wheel    66 Mar 30 15:44 jail-runtime -> /packages/db/jail-runtime-
x86-32-20171012.356211_builder_stable_10
lrwxr-xr-x  1 root  wheel    62 Mar 30 15:44 jdocs -> /packages/db/jdocs-
x86-32-20171121.225603_builder_junos_161_r6
lrwxr-xr-x  1 root  wheel    63 Mar 30 15:44 jpfe-X -> /packages/db/jpfe-X-
x86-32-20171121.225603_builder_junos_161_r6

...
```

To see if there are previous packages on the device, do one of the following:

- From a UNIX shell, issue the `ls /packages/sets/previous` command.
- From the CLI operational mode, use the file `list /packages/sets/previous` command.

Booting from a previously installed release of Junos OS with upgraded FreeBSD is the first option on the Boot Menu.

System boots the previous Junos OS with upgraded FreeBSD image. If there is no previous image, system boots from the currently installed image.

Boot into Single-User Mode

Single-user mode is a mode in which a multi-user computer operating system boots into a single superuser. It is mainly used for maintenance of multi-user environments.

For devices running Junos OS with upgraded FreeBSD, single-user mode puts you in a shell with a prompt. There is limited support and password recovery is not possible using this option. But you can do a few file operations.

Booting into single-user mode is the second option on the Boot Menu..

Boot from a Recovery Snapshot

A recovery snapshot for devices running Junos OS with upgraded FreeBSD is taken with the `request system snapshot recovery` command. Recovery snapshots are full copies of the packages and configuration taken at the time the snapshot command is issued.

Booting from a recovery snapshot is the third option on the Boot Menu.

Boot from a USB Device

If you want to boot from a USB device, you must connect the USB device to the device. Then select the **Boot from [U]SB** option on the Boot Menu. If no USB device is connected, you will see a message No USB media found.



NOTE: On Linux-based platforms (**jinstall-host*** images) where Junos OS with upgraded FreeBSD runs as a guest virtual machine (VM), the boot from USB option is supported through the BIOS Boot Manager. After rebooting, press **ESC** to open the boot options menu and select the **Boot Manager** option.



NOTE: For SRX300, SRX320, SRX340, SRX345, and SRX380 firewalls, you cannot use the Boot menu to boot from a USB device.

Boot to the OAM Shell

The Boot to the OAM Shell option is similar to the single-user mode except that you are put into the oam shell or volume. The compact flash drive is the /oam volume and stores recovery snapshot backup information. In case of failure of the /junos volume, the /oam volume can be used to boot the system.

Booting to the oam shell is the fifth option on the Boot Menu.

CLI Recovery Mode

If you choose the CLI Recovery Mode option, you end up at a root> prompt. Enter `configure` at the prompt to enter the configuration CLI mode. From there you can change the root password to recover your access to the device (see [Recovering the Root Password on Routers](#)).

The CLI Recovery Mode is the second option on the Options Menu.

Check File System

The check file system option lets you make sure there are no issues or corrupted files. The system boots from the OAM volume to perform disk checks. This is the third option on the Options Menu.

Enable/Disable Verbose Boot

Choosing the fourth option on the Options Menu either enables verbose boot, which lets you see the whole boot scroll by, or disables verbose boot.

Boot Prompt

The Boot Prompt option displays an OK prompt from which you can type one of the following commands:

- `menu`—Takes you back to the Junos Main Menu.
- `boot-junos`—Boots the device to the current version of Junos OS.
- `reboot`—Reboots the system.

You can also type `?` at the OK prompt to see several other available commands. The boot prompt option is the fifth option on the Options Menu.

How to Access the Junos Main Menu, Boot Menu, and Options Menu

SUMMARY

If a device running Junos OS with upgraded FreeBSD has a damaged operating system or configuration, preventing the system from booting normally, you can still boot using an option on the Junos Main Menu, Boot Menu, or Options Menu.

IN THIS SECTION

- [How to Access the Junos Main Menu | 347](#)
- [How to Access the Boot Menu | 349](#)
- [How to Access the Options Menu | 350](#)

The following procedures show you how to access these menus.

How to Access the Junos Main Menu

You access the Junos Main Menu by interrupting the reboot of a device.



NOTE: You need console access (either direct access to console or via a console server) to perform the following procedure.

You can either perform the entire procedure or power-cycle the device and start the procedure from Step 4. (You can also perform these reboots by rebooting the device via the CLI if that is available.)

To boot a device running Junos OS with upgraded FreeBSD without using the CLI:

1. Power off the device, such as a router or a switch, by pressing the power button on the front panel.
2. Connect and configure the management device, such as a PC or a laptop, as follows:
 - a. Turn off the power to the management device.

- b. Plug one end of the Ethernet rollover cable supplied with the device into the RJ-45-to-DB-9 serial port adapter supplied with the device.
- c. Plug the RJ-45-to-DB-9 serial port adapter into the serial port on the management device.
- d. Connect the other end of the Ethernet rollover cable to the console port on the device.
- e. Turn on the power to the management device.
- f. On the management device, start your asynchronous terminal emulation application (such as Microsoft Windows Hyperterminal) and select the appropriate communication (COM) port to use (for example, COM1).
- g. Configure the port settings as follows:
 - Bits per second: 9600
 - Data bits: 8
 - Parity: None
 - Stop bits: 1
 - Flow control: None

3. Power on the device by pressing the power button on the front panel.

Verify that the **POWER** LED on the front panel turns green.

The terminal emulation screen on your management device displays the boot sequence of the device.

4. Access the Junos Main Menu.

Do one of the following

- Prior to Junos OS Release 17.3, the Junos Main Menu appears for three seconds on startup before automatically booting from the **/junos** volume. Press any key within the three-second interval to stop the automatic boot sequence and display the Junos Main Menu.
- Starting in Junos OS Release 17.3, press Ctrl+c at the following part in the reboot:

```
FreeBSD/x86 bootstrap loader, Revision 1.1
(builder@feyrith.juniper.net, Sun Feb  4 13:06:24 PST 2018)
/
Autoboot in 1 seconds... (press Ctrl-C to interrupt)
```

The Junos Main Menu is displayed:

```
Main Menu

1.  Boot [D]irectly from Junos volume
2.  Boot Junos volume in [S]afe mode

3.  [R]eboot

4.  [B]oot menu
5.  [M]ore options

Choice:
```

5. At the Choice: prompt in the Junos Main Menu, enter the number representing the option you want to use. Alternatively, you can enter the letter in square brackets to choose an option.

How to Access the Boot Menu

The Boot Menu is one of two menus you can access from the Junos Main Menu.



NOTE: You need console access to perform the following procedure.

You must first access the Junos Main Menu. See ["How to Access the Junos Main Menu, Boot Menu, and Options Menu" on page 347](#).

To access the Boot Menu:

1. At the Choice: prompt in the Junos Main Menu, enter **4** or **B** to choose 4. [B]oot menu.

The Boot Menu is displayed. Only options that are available to your device will appear in the menu, and the menu numbers will change based on the available options.

```
Boot Menu

1.  Boot [P]revious installed Junos packages
2.  Boot [A]ctive Junos packages

3.  Boot Junos in [S]ingle user mode

4.  Boot from [R]ecovery snapshot
5.  Boot from [N]etwork
```

```

6. Boot from [U]SB

7. Boot to [O]AM shell

8. Snapshot [B]oot menu

9. [M]ain menu

Choice:

```

2. At the Choice: prompt in the Boot Menu, enter the number representing the option you want to use. Alternatively, you can enter the letter in square brackets to choose an option.

How to Access the Options Menu

The Options Menu is one of two menus you can access from the Junos Main Menu.



NOTE: You need console access to perform the following procedure.

You must first access the Junos Main Menu. See ["How to Access the Junos Main Menu, Boot Menu, and Options Menu" on page 347](#).

To access the Options Menu:

1. At the Choice: prompt in the Junos Main Menu, enter **5** or **M** to choose 5. [M]ore options.

The Options Menu is displayed.

```

Options Menu 4

1. Recover [J]unos volume
2. Recovery mode - [C]LI

3. Check [F]ile system

4. Enable [V]erbose boot

5. [B]oot prompt

6. [M]ain menu

Choice:

```

2. At the `Choice:` prompt in the Options Menu, enter the number representing the option you want to use. Alternatively, you can enter the letter in square brackets to choose an option.

Rescue and Recovery of Configuration File (Junos OS)

IN THIS SECTION

- [Saving and Reverting a Rescue Configuration File | 351](#)
- [Copy Backup Configurations and Restore Saved Configurations | 356](#)
- [Reverting to the Default Factory Configuration by Using the `request system zeroize` Command | 359](#)

In the event of software failure, a rescue configuration helps to load a known working configuration. No need to remember the rollback number; if you saved a configuration, you can use it anytime when needed.

Saving and Reverting a Rescue Configuration File

IN THIS SECTION

- [Saving a Rescue Configuration File | 351](#)
- [Reverting to the Rescue Configuration | 355](#)

Saving a Rescue Configuration File

IN THIS SECTION

- [Saving a Rescue Configuration | 353](#)
- [Validating the Rescue Configuration | 353](#)
- [Copying the Configuration to a Remote Server | 353](#)
- [Rolling Back to Troubleshoot the Failed Configuration | 354](#)

- [Rolling Back to the Rescue Configuration | 355](#)
- [Deleting an Existing Rescue Configuration | 355](#)

A rescue configuration file is helpful in the event that your device's configuration file has been misconfigured. A rescue configuration allows you to define a known working configuration or a configuration with a known state that you can roll back to at any time. This alleviates the necessity of having to remember the rollback number with the rollback command. You can restore the device to this rescue configuration to bring the device back online. If you save this file off the device, the rescue configuration can also be used to restore your device in the event of a software failure.

For devices running Junos OS with upgraded FreeBSD, provided you have saved a rescue configuration on the device, there is an automatic device recovery mode that goes into action should the system fail to activate the current configuration (amnesiac mode).



NOTE: To determine which platforms run Junos OS with upgraded FreeBSD, see [Feature Explorer](#), enter `freebsd`, and select Junos kernel upgrade to FreeBSD 10+.

You can identify that the device has recovered automatically from amnesiac mode by the following:

- The syslog `UI_DEVICE_IN_RECOVERY_MODE` is generated, which indicates that there was a problem in the normal boot time commit and that Junos OS has activated the rescue configuration as the device's configuration.
- The CLI displays the banner `Device is running in Recovery Mode` in both operational and configuration modes.

Starting in Junos OS Release 23.4R1 for MX Series routers, you can also prevent the device from reaching an amnesiac state post-reboot by configuring the `dual-phase-bootup` feature before the reboot. When a device has a scaled configuration or has a lot of constraints to be validated, upon reboot it may take more than 45 minutes to finish. This lengthy reboot time exceeds the limit set for the watchdog timer. The watchdog timer going off can cause the device to reach an amnesiac state. To avoid reaching an amnesiac state during a future reboot, configure the `dual-phase-bootup` statement.

If you have configured the `dual-phase-bootup` statement before the reboot, the device picks up the rescue configuration from the next reboot. Post-reboot, the device's operational state is active and the device automatically loads the last-configured user configuration (**juniper.conf** file), thus preventing the device from reaching an amnesiac state.

To be able to commit the configuration for the `dual-phase-bootup` statement, you must already have created a rescue configuration (**rescue.conf** file). We recommend that you have a minimal rescue configuration.

This topic covers the following procedures:

Saving a Rescue Configuration

To save a current device configuration as a rescue configuration file:

1. Edit the configuration file on the device to reflect the base configuration you wish to use.
2. In the CLI operational mode, save this edited base configuration as the rescue configuration file:

```
user@host> request system configuration rescue save
```

The rescue configuration file is automatically saved under **/config** directory as **rescue.conf.gz**.

Validating the Rescue Configuration

You can verify that the syntax of a configuration file is correct and check for commit check errors by using the `test configuration filename` command.

To verify if a rescue configuration file is correct:

- Issue the `test configuration filename` command from the CLI operational mode.

```
user@host> test configuration /config/rescue.conf.gz  
configuration check succeeds
```

If the configuration contains any syntax or commit check errors, a message is displayed to indicate the line number and column number in which the error was found. This command only accepts text files.

Copying the Configuration to a Remote Server

This task is optional but recommended.

To copy the rescue configuration to a remote server:

1. Start the device shell.

```
user@host> start shell
```

2. Go to the **/config** directory and list the rescue configuration file..

```
% cd /config  
% ls -lrt rescue.conf.gz  
-rw-r----- 1 root  wheel  1483 Dec 14 10:50 rescue.conf.gz
```

3. FTP the configuration file to the remote host.

```
% ftp host2
Name: username
Password: password
User user logged in.
ftp> cd /var/tmp
ftp> lcd /config
ftp> bin
ftp> put rescue.conf.gz
local: rescue.conf.gz remote: rescue.conf.gz

Transfer complete.
ftp> bye
Goodbye.
```

Rolling Back to Troubleshoot the Failed Configuration

Your rescue configuration is probably not exactly the configuration you want or need on your system. Therefore, you will want to examine the failures that occurred when you tried to activate the current configuration and make corrective actions.

To correct the failed configuration:

1. Log in to the device through the management IP (or the console if permitted).
2. Load the failed configuration.

```
user@host# rollback 1
```

If you are doing this step right after the recovery mode, `rollback 1` will be the configuration that cause the amnesiac mode.

3. Make corrections to the configuration.
4. Do a commit check.

```
user@host# commit check
```

5. If there are other corrections to make, make them.
6. Commit the configuration.

Rolling Back to the Rescue Configuration

Not all platforms run Junos OS with updated FreeBSD. Those that do not or are releases earlier than Junos OS Release 16.1, do not have the automatic recovery mode. You will need to rollback to rescue configuration manually to bring the device back to normal running mode.

To roll back to the rescue configuration:

1. Log in to the device through the console.
2. Issue the `rollback rescue` command from the configuration mode of the CLI.

```
user@host# rollback rescue
```

```
load complete
```

3. Commit the configuration.

```
user@host# commit
```

4. Fix the failed configuration. See ["Rolling Back to Troubleshoot the Failed Configuration" on page 354](#).

Deleting an Existing Rescue Configuration

To delete an existing rescue configuration:

- Issue the `request system configuration rescue delete` command:

```
user@host> request system configuration rescue delete
```

Reverting to the Rescue Configuration

If someone inadvertently commits a configuration that denies management access to a device and the console port is not accessible, you can overwrite the invalid configuration and replace it with the rescue configuration. The rescue configuration is a previously committed, valid configuration.

To revert the switch to the rescue configuration:

1. Enter the `load override` command.

```
[edit]  
user@host# load override filename
```

2. Commit your changes.

```
[edit]
user@host# commit filename
```

Copy Backup Configurations and Restore Saved Configurations

IN THIS SECTION

- [Copy Backup Configurations to the Router | 356](#)
- [Restoring a Saved Configuration | 357](#)

Copy Backup Configurations to the Router

To copy backup configurations to the router, follow these steps:

1. To copy the existing configuration and any backup configurations back onto the router, use the `file copy` command. Place the files in the `/var/tmp` directory.

```
user@host> file copy var/tmp/filename
```

2. Load and activate the desired configuration:

```
user@host> configure
[edit]
user@host# load merge/config/filename or load replace/config/
filename
[edit]
user@host# commit
```

Restoring a Saved Configuration

IN THIS SECTION

- [Copy Saved Files to the Router | 357](#)
- [Loading and Committing the Configuration File | 358](#)

To restore a saved configuration, perform the following tasks:

Copy Saved Files to the Router

To copy the saved configuration to the router:

1. Log in to the console as root. There is no password.

```
Escape character is '^['.
```

```
[Enter]
```

```
router (ttyd0)
```

```
login: root
```

```
Password: [Enter]
```

Initially, access to the router is limited to the console port after a recovery installation. Access through the management ports and interfaces is set in the configuration. For information about accessing the router through the console port, see the administration guide for your particular router.

2. Start the CLI:

```
# cli
```

3. Copy the configuration file on the remote server to the router's **/var/tmp** directory:

```
root@host> ftp remote-server
```

```
user: username
```

```
password: password
```

```
ftp> bin
```

```
Type set to I.
```

```
ftp> get /path/file
```

```
ftp> bye
Goodbye.
```

Loading and Committing the Configuration File

Once the saved configuration file is copied to the router, you load and commit the file:

1. Start the CLI configuration mode.

```
user@host> configure
Entering configuration mode

[edit]
user@host#
```

2. Load the file into the current configuration. You should override the existing file.

```
user@host#
load override /var/tmp/filename
load complete
```

3. Commit the file.

```
user@host# commit
commit complete
```

4. Exit the CLI configuration mode.

```
user@host# exit
user@host>
```

5. Back up Junos OS.

After you have installed the software on the router, committed the configuration, and are satisfied that the new configuration is successfully running, issue the `request system snapshot` command to back up the new software to the `/altconfig` file system. If you do not issue the `request system snapshot` command, the configuration on the alternate boot drive will be out of sync with the configuration on the primary boot drive.

The `request system snapshot` command causes the root file system to be backed up to `/altroot`, and `/config` to be backed up to `/altconfig`. The root and `/config` file systems are on the router's CompactFlash card, and the `/altroot` and `/altconfig` file systems are on the router's hard disk or solid-state drive (SSD).

Reverting to the Default Factory Configuration by Using the `request system zeroize` Command

The `request system zeroize` command is a standard Junos OS operational mode command that removes all configuration information and resets all key values. The operation unlinks all user-created data files, including customized configuration and log files, from their directories. The device then reboots and reverts to the factory-default configuration.

To completely erase user-created data so that it is unrecoverable, use the `request system zeroize media` command.



CAUTION: Before issuing `request system zeroize`, use the `request system snapshot` command to back up the files currently used to run the device to a secondary device.

To revert to the factory-default configuration by using the `request system zeroize` command:

1. Remove the device from the chassis cluster.
2. Disable the chassis cluster on the device.
3. Reboot the device.
4. Enter the `request system zeroize` command.

```
user@host> request system zeroize
warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files? [yes,no] (yes)
```

5. Type **yes** to remove configuration and log files and revert to the factory default configuration.
6. Complete the initial configuration of the device.

SEE ALSO

request system zeroize (Junos OS)

Autorecovery of Configuration, Licenses, and Disk Information on SRX Series Devices

SUMMARY

Autorecovery helps to detect and recover information on disk partitioning, configuration, and licenses if the disk becomes corrupted.

IN THIS SECTION

- [Overview | 360](#)
- [How Autorecovery Works | 360](#)
- [How to Use Autorecovery | 361](#)
- [Data That Is Backed Up in an Autorecovery | 361](#)
- [Troubleshooting Alarms | 362](#)
- [Considerations | 362](#)

Overview

The autorecovery feature is supported on dual-partitioned SRX Series Firewalls. With this feature, information on disk partitioning, configuration, and licenses is recovered automatically in the event it becomes corrupted.

In devices running FreeBSD Release 12 or later, you cannot back up data with the autorecovery feature. Instead, back up data using snapshots. To learn if your device is running FreeBSD Release 12 or later, issue the `show version` command and look for the `fbbsd_builder_stable` string in the module names. If the string includes the number 12 or later, your device is running FreeBSD Release 12 or later.

Autorecovery provides the following functions:

- Detects corruption in disk partitioning during system bootup and attempts to recover partitions automatically.
- Detects corruption in the Junos OS rescue configuration during system bootup and attempts to recover the rescue configuration automatically.
- Detects corruption in Junos OS licenses during system bootup and attempts to recover licenses automatically.

How Autorecovery Works

The feature works in the following ways:

- The `request system autorecovery state save` command backs up important data such as disk partitioning information, licenses, and Junos OS rescue configuration.
- Once the backup copies are saved, the copies are used to check the integrity of the working copies of the data on every bootup.
- The working copies are automatically recovered if any corruption is detected.

How to Use Autorecovery

You use autorecovery in the following ways:

- Prepare the router for deployment with the necessary licenses and configuration.
- After you finalize the state, issue the `request system autorecovery state save` command to back up the state.
- After you save the state, integrity check and recovery actions (if any) occur automatically on every bootup.
- If subsequent maintenance activities change the state of the router by adding licenses or updating the configuration, you need to issue the `request system autorecovery state save` command again to update the saved state.
- Issue the `show system autorecovery state` command any time to view the status of the saved information and the integrity check status of each saved item.
- Issue the `request system autorecovery state clear` command to delete all backed up data and disable autorecovery, if required.

Data That Is Backed Up in an Autorecovery

The following data is backed up during the autorecovery process:

- Rescue configuration (regenerated from the current configuration)
- License keys
- BSD labels (disk-partitioning information)

Data is backed up only when you issue the `request system autorecovery state save` command. Disk-partitioning information is backed up automatically from factory defaults (for new systems), on installation from the boot loader, and on snapshot creation.

Troubleshooting Alarms

Table 19 on page 362 lists types of autorecovery alarms, descriptions, and required actions.

Table 19: Autorecovery Alarms

Alarm	Alarm Type	Description	Action Required
Autorecovery information needs to be saved	Minor	This alarm indicates: <ul style="list-style-type: none"> Unsaved data needs to be saved, or saved data contains problems and another save is required. 	<ul style="list-style-type: none"> Ensure that the system has all required licenses and configuration. Issue the request system autorecovery state save command. <p>NOTE: In FreeBSD Release 12 and later, the autorecovery feature does not support data backup. Use alternative backup methods</p>
Autorecovery has recovered corrupted information	Minor	This alarm indicates: <ul style="list-style-type: none"> Boot time integrity check failed for certain items; however, the items have been recovered successfully. 	<ul style="list-style-type: none"> No action is required. Alarm is cleared on next bootup.
Autorecovery was unable to recover data completely	Major	This alarm indicates: <ul style="list-style-type: none"> Boot time integrity check failed for certain items, which could not be recovered successfully. 	<ul style="list-style-type: none"> The system might be experiencing a fatal malfunction.

Considerations

- Devices must have dual-root partitioning for autorecovery to work.
- The request system configuration rescue save command regenerates the rescue configuration from the current Junos OS configuration and then saves it. Therefore, issuing the save command overwrites any existing rescue configuration.

- In general, the saved contents of the rescue configuration are not updated automatically. If you add licenses, you must issue the `request system autorecovery state save` command again.

The rescue configuration is backed up. If **/config** is corrupted, the system boots from the rescue configuration.

RELATED DOCUMENTATION

[Create a Snapshot and Using It to Boot an SRX Series Firewall | 318](#)

Example: Installing Junos OS Upgrade Packages on SRX Series Devices

Reverting the Junos OS Software Image Back to the Previous Version

6

PART

Install, Upgrade, and Downgrade VM Host Software, and Back Up and Recover VM Host Devices

- VM Host Overview and CLI Commands | **365**
 - Boot Process for Routers with VM Host Support | **382**
 - Install, Upgrade, Back Up, and Recover VM Host | **385**
 - Copy the VM Host Installation Package to the PXE Boot Server | **392**
 - Upgrade NFX Devices | **396**
 - Create an Emergency Boot Device for Routing Engines with VM Host Support | **416**
 - Upgrade Firmware on Routing Engines with VM Host Support | **418**
 - Disable Autorecovery on Routing Engines with VM Host Support | **433**
 - Personality Upgrade | **434**
-

VM Host Overview and CLI Commands

IN THIS CHAPTER

- VM Host Overview (Junos OS) | 365
- VM Host Operations and Management | 380

VM Host Overview (Junos OS)

IN THIS SECTION

- What Are VM Hosts? | 365
- Routing Engines with VM Host Support | 367
- Salient Features of the Routing Engines with VM Host Support | 372
- Enhanced VM Host Architecture | 379

What Are VM Hosts?

Starting in Junos OS Release 16.1, virtualized Routing Engines are supported that not only provide increased control plane scalability and performance but also provide virtualization capabilities to the Junos OS infrastructure. These virtualized Routing Engines, or VM hosts, are listed in [Hardware Specifications of the Routing Engines with VM Host Support](#).



NOTE: VM hosts only run Junos OS with Upgraded FreeBSD.

The rest of this section describes the architecture of VM hosts. For more information on VM hosts, see the chapters on System Back Up and Recovery, Installing Software, Installing Firmware, and so on in this guide.

[Figure 13 on page 367](#) illustrates the architecture of Routing Engines with VM Host support. It comprises the following components:

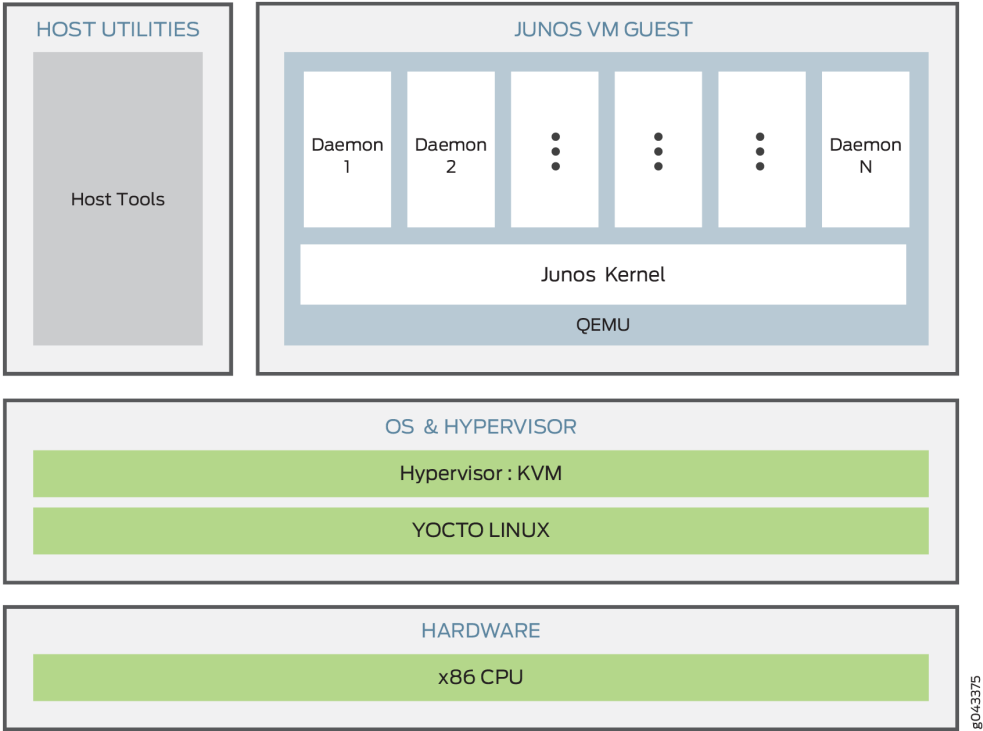
- The hardware layer
- The operating system and hypervisor layer.
- The host utilities and Junos VM guest layer.

The server at the hardware layer contains the physical network interface cards (NICs), CPUs, memory, and Ethernet management port. The NICs support hardware virtualization based on single root I/O virtualization (SR-IOV). With SR-IOV, the physical NICs (known as a physical functions) are managed by the host, while the virtual functions are managed by the guest OS. Over the hardware layer, a Linux-based OS provides the host environment along with the kernel-based virtual machine (KVM) and Quick Emulator (QEMU). This host OS manages the boot complex, CPU memory storage, and various other hardware components such as the physical functions. Junos OS runs as guest OS, manages the virtual functions, and serves as the administrative framework. Additionally, it also provides the interface for managing the host and the hypervisor.

The additional applications and utilities running on the host OS assist in providing the following functionality:

- Facilitating communication between host OS and guest OS.
- Triggering appropriate execution of the host OS based on the command and configuration on the guest Junos OS.
- Extending the VM management functionality to provide features such as autorecovery.

Figure 13: Architecture of Routing Engines with VM Host Support



Routing Engines with VM Host Support

The Routing Engines with VM host support not only provide increased control plane scalability and performance but also provide virtualization capabilities to the Junos OS infrastructure to support greater computing demands.

Virtualization enables multiple instances of operating systems, called guests, to run concurrently on the host and share virtualized hardware resources. A guest is a virtual machine (VM) that runs on a hypervisor-based host and shares its resources. A host is a virtualized software whose hypervisor allows multiple guest VMs to run on it concurrently and share its resources. The VMs must be instances of Junos OS. Third-party VMs are not supported on these Routing Engines. Each VM runs its own operating system image and applications that can be different from that of another VM running on the same host.



NOTE: Only Junos OS VM are supported. You cannot run third party VMs on these Routing Engines.

On the Routing Engines with VM host support, one instance of Junos OS runs as a VM over a Linux-based host (VM host) and serves as the VM operating in the administrative context. Junos OS manages

all configurations, chassis control, communication with the host OS, and user interface command execution, thus providing near-native Junos OS experience to the end user.

See [Table 20 on page 368](#) for more information on hardware specifications of the Routing Engines with VMHost support.

Table 20: Hardware Specifications of the Routing Engines with VM Host Support

Model Number	Supported on Device	Specifications
RE-ACX-5448	ACX5448	<ul style="list-style-type: none"> • High-performance 1.6-GHz Intel 8 Core X86 CPU • 32-GB two DIMM DRAM • Two 100-GB SATA SSD
EX9200-RE2	EX9204, EX9208, and EX9214	<ul style="list-style-type: none"> • Six-core, 2-GHz Intel processor • 64-GB of DRAM and dual front pluggable SSDs, each providing 64-GB of storage for Junos OS images and logs.
RE-S-1600x8	MX204	<ul style="list-style-type: none"> • High-performance 1.6-GHz Intel 8 Core X86 CPU • 32-GB DDR4 RAM • 100-GB SATA SSD
RE-S-X6-64G	MX240, MX480, and MX960	<ul style="list-style-type: none"> • 6-core Haswell CPU • Wellsburg PCH-based Routing Engine with 64-GB DRAM and two 64-GB solid-state drives (SSDs)

Table 20: Hardware Specifications of the Routing Engines with VM Host Support (*Continued*)

Model Number	Supported on Device	Specifications
RE-S-X6-128G	MX240, MX480, and MX960	<ul style="list-style-type: none"> • 6-core Haswell CPU • Wellsburg PCH-based Routing Engine with 128-GB DRAM and two 128-GB solid-state drives (SSDs)
REMX2008-X8-64G-LT,	MX2008	<ul style="list-style-type: none"> • 8-core Haswell CPU • Wellsburg PCH-based Routing Engine with 64-GB DRAM and two 100-GB solid-state drives (SSDs)
REMX2008-X8-128G-S		<ul style="list-style-type: none"> • 8-core Haswell CPU • Wellsburg PCH-based Routing Engine with 128-GB DRAM and two 200-GB solid-state drives (SSDs)
REMX2K-X8-64G	MX2020 and MX2010	<ul style="list-style-type: none"> • 8-core Haswell CPU • Wellsburg PCH-based Routing Engine with 64-GB DRAM and two 64-GB SSDs
RE-S-1600x8	MX10003	<ul style="list-style-type: none"> • High-performance 1.6-GHz Intel 8 Core X86 CPU • 64-GB DDR4 RAM • 100-GB SATA SSD

Table 20: Hardware Specifications of the Routing Engines with VM Host Support (*Continued*)

Model Number	Supported on Device	Specifications
JNP10K-RE1, JNP10K-RE1-LT, and JNP10K-RE1-128	MX10008 MX10004	<ul style="list-style-type: none"> • High-performance 2.2-GHz Intel 10 Core X86 CPU • 64-GB DDR4 RAM • Two 200-GB SATA SSD
JNP304-RE-S	MX304	<ul style="list-style-type: none"> • 8-core, Intel Icelake Based Multicore Processor CPU • 128-GB of DRAM • Two 200-GB SATA SSD
RCBPTX	PTX3000	<ul style="list-style-type: none"> • Wellsburg PCH-based Routing Engine with 64-GB DRAM and two 64-GB SSDs • Multi-core Haswell CPU <p>RCB combines the functionality of a Routing Engine, Control Board, and Centralized Clock Generator (CCG)</p>
RE-PTX-X8-64G	PTX5000	<ul style="list-style-type: none"> • 8-core Haswell CPU • Wellsburg PCH-based Routing Engine with 64-GB DRAM and two 64-GB SSDs • New Control Board CB2-PTX
RE-PTX10002-60C	PTX10002-60C	<ul style="list-style-type: none"> • High-performance 1.6-GHz Intel 8 Core X86 CPU • 32-GB DDR4 RAM • Two 50-GB SATA SSD

Table 20: Hardware Specifications of the Routing Engines with VM Host Support (*Continued*)

Model Number	Supported on Device	Specifications
RE-QFX10002-60C	QFX10002-60C	<ul style="list-style-type: none"> • High-performance 1.6-GHz Intel 8 Core X86 CPU • 32-GB DDR4 RAM • Two 50-GB SATA SSD
SRX5K-RE3	SRX5000	<ul style="list-style-type: none"> • 6-core Haswell CPU • 128-GB of DRAM • Two 128-GB solid-state drives (SSDs)
SRX1600	SRX1600	<ul style="list-style-type: none"> • 4-core, 2.2GHz, Intel Icelake Based Processor CPU • 120-GB NVMe SSD
SRX2300, SRX4120	SRX2300, SRX4120	<ul style="list-style-type: none"> • 8-core, 2.7 GHz, Intel Icelake Based Processor CPU • Two 120-GB NVMe solid-state drives (SSDs)
SRX4300	SRX4300	<ul style="list-style-type: none"> • 20-core, 2.1 GHz, Intel Icelake Based Processor CPU • 120-GB and 960 GB NVMe solid-state drives (SSDs)



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

SEE ALSO

| [Supported Routing Engines by Router](#)

Salient Features of the Routing Engines with VM Host Support

IN THIS SECTION

- [Platform Virtualization | 372](#)
- [Hardware Assisted Paravirtualized Guest Junos OS | 372](#)
- [Guest Junos OS to Serve as the Administrative Framework | 373](#)
- [Storage Partitioning and Redundancy | 373](#)
- [NTP and Time Zone | 377](#)
- [Autorecovery | 377](#)
- [Handling Reboot and Power Off | 377](#)

While continuing to provide the same end-user experience, the new architecture provides a better performing Routing Engine.

The following are the salient features of the Routing Engines:

Platform Virtualization

Platform virtualization by the introduction of a middle layer that comprises the host OS and the KVM (or the hypervisor).

- Enables support for multiple instances of Junos OS to be run concurrently.
- Enables support for third-party software to be run directly.

Hardware Assisted Paravirtualized Guest Junos OS

Provides the user with the benefits of platform virtualization along with the default performance and functionality. Paravirtualization is a virtualization technique in which a software component similar to the underlying hardware component resides in the VM and interacts with the hypervisor to execute many operations. In contrast to full virtualization, this technique reduces the overhead of virtualization in the VM.

Guest Junos OS to Serve as the Administrative Framework

The configurations, chassis control, communication with the host OS, and user interface command execution are managed by the guest Junos OS.

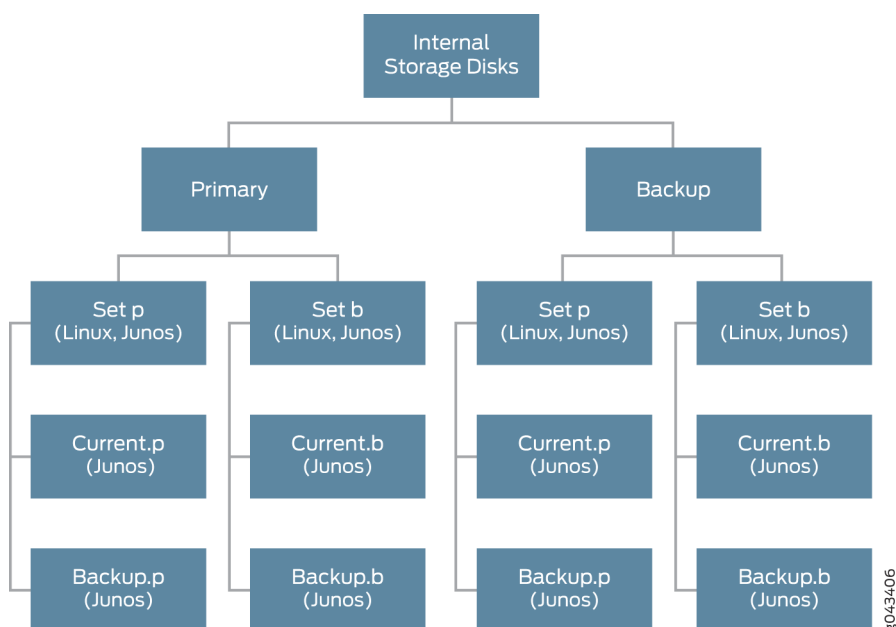
Storage Partitioning and Redundancy

An Internal solid-state drive (SSD) is used as boot media for operating the Routing Engine. Additional options such as USB storage and network boot are available for installation and recovery purposes. A set of two 50-GB SSDs is available for normal functioning of the Routing Engine. The Routing Engine requires both the SSDs to be functional. Storage partitioning is important for debugging the Routing Engine, for new installations, and for SSD replacement.

Of the two SSDs, one operates as the primary SSD and the other as the backup SSD. Two sets of software boot images—the current set and the alternate (or previous) set are available on the primary SSD. The system boots from the current set, while the alternate set contains the previous version of the software boot image. After a software upgrade, the new version of the software is available on the alternate set. When the device is rebooted after the upgrade, the alternate set becomes the new current set and the current set, which now carries an older version of the software image, becomes the alternate set. You can switch to alternate set by using the `request vmhost software rollback` command. Until a software upgrade or a software rollback is performed, the system is programmed to boot from the same set of images on the disk.

Both the SSDs are partitioned to provide host boot partition, root partition, and partition for the guest image storage. The host boot partition contains the boot loader, which is the software responsible for booting the OS, Linux kernel, and RAM file system. The root partition contains the root file system for the host OS.

[Figure 14 on page 374](#) shows the partitioning of SSDs.

Figure 14: SSD Partitioning

Each SSD partition contains more than one set of fully functional host software. In case of a boot failure on the primary SSD, the router can boot by using the snapshot available on the alternate SSD. This snapshot can be generated by a fresh installation or by using the request `vmhost snapshot` command.

Starting in Junos OS Release 18.1R1, the Routing Engines on the MX240, MX480, MX960, MX2010, MX2020, and PTX5000 support Secure Boot.

Starting in Junos OS Release 18.2R1, the Routing Engine on the MX2008 supports Secure Boot.

The Routing Engines with Secure Boot support have both RAM and SSD upgraded to 128GB and 2x200GB respectively. The increased SSD size facilitates increased storage of core and log files.

The following table provides information on the SSD size for different Routing Engines:

Table 21: SSD Size of Routing Engines

Devices	Routing Engine model number	SSD size
ACX5448	RE-ACX-5448	2x100GB
EX9204, EX9208, and EX9214	EX9200-RE2	2x64GB
MX204	RE-S-1600x8	2x50GB

MX240, MX480, and MX960	RE-S-2200X6-64G-S	2x50GB
	RE-S-X6-64G-LT	2x50GB
	RE-S-X6-128G-S	2x200GB
MX2008	REMX2008-X8-64G-LT	2x100GB
	REMX2008-X8-128G-S	2x200GB
MX2010 and MX2020	RE-MX2K-X8-64G	2x100GB
	RE-MX2K-X8-64G-LT	2x100GB
	RE-MX2K-X8-128G-S	2x200GB
MX10003	RE-S-1600x8	2x50GB
MX10008 MX10004	JNP10K-RE1, JNP10K-RE1-LT, and JNP10K-RE1-128	2x200GB
PTX3000	RCBPTX	2x64GB
PTX5000	RE-PTX-X8-64G	2x64GB
PTX10002-60C	RE-PTX10002-60C	2x50GB
QFX10002-60C	RE-QFX10002-60C	2x50GB
SRX5000	SRX5K-RE3	2x128GB

You can use the `show vmhost hardware` command to display the increased RAM size, SSD size, and other hardware information.

The following illustration explains the partition of the host to facilitate the increased storage of core files and log files. [Figure 15 on page 376](#) illustrates the partition of the host on MX240, MX480, MX960,

MX2008, and PTX5000 routers with the 200-GB SSDs. A virtual disk of size 56-GB will be allocated from VM partition to the guest as var-config.disk. The current size of this disk is 15-GB.

Figure 15: Host partition table for Routing Engines with 200-GB SSDs



Figure 16 on page 376 illustrates the storage allocation of the guest VM.

Figure 16: Partitioning of the guest VM



NOTE: For Routing Engines with 50GB SSD, the host partition remains as-is.

Figure 17 on page 376 and Figure 18 on page 376 illustrates the host partition table and the storage allocation of the guest VM for the MX2010 and MX2020 routers respectively.

Figure 17: Host partition table for Routing Engines on MX2010 and MX2020 routers with 100GB SSD



A virtual disk of size 32-GB is allocated from VM partition to the guest Junos OS as var-config.disk.

Figure 18: Guest VM partition on MX2010 and MX2020 Routers



A reformatting of the SSD is required to implement the enhancement of the /var size. The upgrade can be implemented by any of the following methods:

- Installation from SSD Disk2-Boot the host OS from the backup disk (SSD Disk2) and install the junos-vmhost-install-x.tgz image.
- Installation from USB

NTP and Time Zone

The date and time zones are synchronized from the administrative guest Junos OS to the host OS. Therefore, the timestamps in system log files of Junos OS and the host OS are synchronized.

Autorecovery

The automatic recovery (autorecovery) feature provides the following functions:

- Detecting corruptions in disk partitioning during system startup and attempting to recover partitions automatically
- Detecting corruptions in the Junos OS configuration during system startup and attempting to recover the configuration automatically, thereby ensuring that the operations and management are not disrupted.
- Detecting corruptions in Junos OS licenses during system startup and attempting to recover licenses automatically.

During the process of recovery, the host OS tries to launch the Junos VM from the image available on the primary disk. However, if the Junos VM fails to launch, the host OS attempts to launch the Junos VM from the snapshot of the host OS image and Junos OS image available in the backup disk, provided `request vmhost snapshot` was the last operation performed. If the backup disk does not contain the snapshot, the host OS attempts to launch the Junos VM from the software available in the alternate set in the primary disk, provided `request vmhost upgrade` was the last operation performed.

The autorecovery feature is enabled by default on the guest OS. If you need to disable autorecovery—for example, to examine the failure state for debugging—use the following command:

```
user@host> set vmhost no-auto-recovery
```

Handling Reboot and Power Off

You can reboot the Routing Engine by using the `request vmhost reboot` command. This command reboots the Routing Engine by rebooting both the guest Junos OS and the host OS. However, reboot of the Routing Engine can be triggered because of various reasons. The events or the reasons that trigger a host OS reboot are different from those that trigger a guest OS reboot.

Guest OS reboot implies that only the Junos OS is rebooted, and that the host OS is up and running. The following are a few of the reasons that trigger a guest OS reboot:

- Reboot due to panic
- VJUNOS reboot—Guest OS reboot after a shutdown.
- VJUNOS watchdog from host—Guest reboot due to emulated watchdog timer expiry

Host OS reboot implies that both the host OS and the guest OS (here, Junos OS) are rebooted. The following are a few reasons that trigger a host OS and guest OS reboot:

- Hypervisor reboot
- Power cycle or power failure
- Reboot due to exception.
- Reset-button reset—Reboot triggered by the pressing of the reset button on the front panel.
- Thermal shutdown
- Watchdog—Reboot due to PCH watchdog timer expiry

You can find the reason for the reboot by using the `show chassis routing-engine 0` command or the `show vmhost uptime` command.

For example:

```
host@router> show chassis routing-engine 0 | match "Last reboot reason"
```

```
Last reboot reason 0x4000:VJUNOS reboot
```

```
host@router> show vmhost uptime re0 | match "Vmhost last reboot reason"
```

```
Vmhost last reboot reason: 0x2000:hypervisor reboot
```

If the Routing Engine finishes booting and if you need to power off the router again, run the `request vmhost power-off` command. If you want the Routing Engine to reboot, use the `request vmhost reboot` command.

Enhanced VM Host Architecture

The integration of TVP and VM host has led to the development of the enhanced VM host architecture, which effectively separates platform-dependent and platform-independent components, as well as guest applications. By maintaining the guest operating system, Junos OS, in a largely platform-independent manner, we enhance both flexibility and performance. To facilitate this, we have transitioned platform-dependent elements to the Linux user space as plugin entities. This structure allows platform and PFE activities to operate independently of Junos OS, thereby increasing overall system efficiency. Furthermore, by utilizing Linux for platform-dependent components, we can also leverage the advantages of open-source software and drivers. For a list of platforms that support the enhanced VM host architecture, see [Feature Explorer](#).

The following lists the main components of the enhanced VMhost architecture:

- Junos VM—A TVP enabled Junos OS based on latest FreeBSD.
- Linux OS—A Wind River based Linux OS will be used as the host operating system
- Storage SSDs—An SSD serves as the internal non-volatile storage for VM host images, logs, and configurations. Every RCB is provisioned with two NVMe SSDs of 400GB each: a primary SSD and a secondary SSD. Each SSD stores two VM host images, referred to as set p and set b.

Change History Table

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

Release	Description
25.2	Starting in Junos OS Release 25.2R1, the MX10004 and MX10008 routers support the enhanced VM host architecture.
18.2	Starting in Junos OS Release 18.2R1, the Routing Engine on the MX2008 supports Secure Boot.

RELATED DOCUMENTATION

<i>request vmhost snapshot</i>
<i>request vmhost reboot</i>
<i>request vmhost power-off</i>

VM Host Operations and Management

With the virtualization of the Routing Engine, Junos OS supports new `request` and `show` commands associated with the host and hypervisor processes. The commands are related to:

- Reboot, halt, and power management for the host.
- Software upgrade for the host.
- Disk snapshot for the host.

The following `request` commands are not available on the RE-MX-X6, RE-MX-X8, and RE-PTX-X8 Routing Engines:

- `request system halt`
- `request system partition abort`
- `request system power-off`
- `request system power on`

The following commands can be used only for the guest Junos OS:

- `request system reboot`
- `request system snapshot`
- `request system software add`
- `request system zeroize`

You can use the following new `request vmhost` commands on the host OS:

- `request vmhost cleanup`
- `request vmhost file-copy`
- `request vmhost halt`
- `request vmhost hard-disk-test`
- `request vmhost power-off`
- `request vmhost power-on`
- `request vmhost reboot`
- `request vmhost snapshot`

- `request vmhost software abort in-service-upgrade`



NOTE: This command is not supported on the QFX10002-60C and PTX10002-60C devices.

- `request vmhost software add`
- `request vmhost software in-service-upgrade`



NOTE: This command is not supported on the QFX10002-60C and PTX10002-60C devices.

- `request vmhost software rollback`
- `request vmhost zeroize`

RELATED DOCUMENTATION

| [Routing Engines with VM Host Support](#) | 367

Boot Process for Routers with VM Host Support

IN THIS CHAPTER

- [Boot Process for Routers with VM Host Support | 382](#)

Boot Process for Routers with VM Host Support

IN THIS SECTION

- [Booting for the First Time | 382](#)
- [Boot Sequence | 382](#)
- [Understanding Console Port | 383](#)
- [Understanding Hostnames Synchronization | 383](#)

The boot process involves configuring the basic parameters through the console port and filename synchronization.

Booting for the First Time

When you power on a device for the first time, the router initiates the boot process.

After hardware and field-programmable gate array (FPGA) level initialization is complete, the Unified Extensible Firmware Interface (UEFI) selects the boot device to launch the host OS. The host OS launches the default guest Junos OS, which is the administrative context for the user. After the device has powered on completely, a login prompt is displayed on the console port.

Boot Sequence

The Routing Engine boots from the storage media in the following sequence:

- USB
- Solid-state Drive 1 (SSD1)
- Solid-state Drive 1 (SSD2)
- Preboot Execution Environment (PXE)

Understanding Console Port

To perform the initial configuration, you need to connect a terminal or laptop computer to the router through the console port, which is a serial port on the front of the router. The console port is the management port used by administrators to log in to Junos OS directly—that is, without using a network connection.

Two universal asynchronous receiver/transmitter (UART) ports are connected to the midplane to provide CTY access to line cards. At any time, two ports can be active for the CTY application. These ports are available to the Junos VMs for configuration.

For more information about configuring the router's basic properties, see [Accessing a Junos OS Device the First Time](#).

Understanding Hostnames Synchronization

A hostname provides a unique identification for a router on the network. Junos OS uses the configured hostname as part of the command prompt, to prepend log files and other accounting information, as well as in other places where knowing the device identity is useful. Although Junos OS supports a maximum hostname length of 255 characters, the host OS supports hostnames that have only 64 characters or less. Therefore, hostnames need to be synchronized between Junos OS and the host OS. Keep in mind the following conditions when you synchronize the hostname configured on Junos OS with that on the host OS:

- If the Junos OS-configured hostname has less than or equal to 58 characters, then the hostname supported by the host OS (Linux) has the format *Junos hostname-node*.

For example, if the Junos OS-configured hostname is *xx.xx*, the hostname is *xx.xx-node*.

- If the Junos OS-configured hostname is greater than 58 characters in length, then the synchronization process truncates characters from the 59th character onward and replaces the truncated characters with *-node*.

RELATED DOCUMENTATION

[Creating an Emergency Boot Device for Routing Engines with VM Host Support](#) | 416

vmhost

request vmhost reboot

request vmhost power-off

[Creating an Emergency Boot Device for Routing Engines with VM Host Support](#) | 416

Install, Upgrade, Back Up, and Recover VM Host

IN THIS CHAPTER

- [Installing, Upgrading, Backing Up, and Recovery of VM Host | 385](#)

Installing, Upgrading, Backing Up, and Recovery of VM Host

IN THIS SECTION

- [VM Host Upgrade | 386](#)
- [VM Host Rollback | 388](#)
- [VM Host Snapshot | 390](#)

You can install the Junos OS software package and host software package on the device. The following installation options are available:



NOTE: The VM Host installation works differently on the QFX10002-60C switch and PTX10002-60C router. See ["Installing Software Packages on QFX Series Devices" on page 174](#) and [Installing the Software on PTX10002-60C Routers](#) for more details. However, the information on the rollback and snapshot features work the same on QFX10002-60C switches and PTX10002-60C routers.

- **Fresh installation**— This installation method can be used for factory installation as well as for recovery after corruption. Fresh installation can be done using Preboot Execution Environment (PXE)/NetBoot or a USB install media package. This method of installation installs the host OS, tools, and the Junos VMs.

A PXE boot is an environment to boot devices using a network interface independent of available data storage devices or installed operating systems. The PXE environment is built on a foundation of

Internet protocols and services. These include TCP/IP, DHCP, and TFTP. This method of installation is mostly used for installing the operating system on a device, without depending on the state of the internal media. The required software for network installation is stored on a TFTP server. PXE boot method supports remote installation thereby overcoming the need for an in-person assistance for installation. For more information, see ["Copying VM Host Installation Package to the PXE Boot Server" on page 392](#). After you copy the VM Host Installation Package to the PXE Boot Server, you can use the `request vmhost reboot network` command and reboot the device to install the software. The device boots from the PXE server and installs the software on both the SSDs.

You can choose to use the USB disk installation method when the device fails to reboot because of internal media failure or when there is no installed Junos OS. For more information, see ["Creating an Emergency Boot Device for Routing Engines with VM Host Support" on page 416](#).

On a fresh installation using USB, the following directories are populated with the Junos OS image on both the SSDs:

- `Current.p`
- `Backup.p`
- `Backup.b`
- Regular installation— This installation method is generally for an upgrade or a downgrade. This procedure can be used to install the runtime installation package on the currently running Junos VM to upgrade or downgrade relevant components. Junos VM performs the dependency check to identify the software components that require an upgrade or a downgrade to ensure compatibility.



NOTE: The RE-S-X6-64G-LT and RE-MX2K-X8-64G-LT Routing Engines are restricted to boot only the Junos OS with upgraded FreeBSD Limited image. They fail to boot if you try to install or upgrade the device with an image other than the Limited image, which begins with the **junos-vmhost-install** prefix.

VM Host Upgrade

Every Junos OS release is a group of files bundled together. The Routing Engines RE-MX-X6, RE-MX-X8, and RE-PTX-X8 support only the 64-bit version of Junos OS.



NOTE: If you have important files in directories other than `/config` and `/var`, copy the files to a secure location before upgrading the device. The files under `/config` and `/var` (except `/var/etc`) are preserved after the VM host upgrade.



NOTE: Before installing software on a device that has one or more custom YANG data models added to it, back up and remove the configuration data corresponding to the custom YANG data models from the active configuration. For more information see ["Managing YANG Packages and Configurations During a Software Upgrade or Downgrade" on page 205.](#)

In order to perform VM Host upgrade, use the **junos-vmhost-install-x.tgz** image. This upgrade installs the host image along with the compatible Junos OS.



NOTE: To upgrade the Junos OS on RE-S-X6, RE-MX-X8, and RE-PTX-X8 Routing Engines, always use the VM Host Installation Package. Do not use the jinstall package.



NOTE: Starting with Junos OS Release 21.4R1 and later, on the ACX5448, MX204, MX240, MX480, MX960, MX2010, MX2020, and MX2008 routers with VM host support, during an upgrade or reboot, the **root** login is required for copying the image from the Junos VM to the Linux host. Before the upgrade, you must delete the system services ssh root-login deny statement or change the configuration to system services ssh root-login deny-password. Once the upgrade is complete, you can add the system service ssh root-login deny statement back to your configuration. See <https://kb.juniper.net/>

The following example illustrates the upgrade operation. You can install multiple software packages and software add-on packages at the same time.

```
user@host> > request vmhost software add /var/tmp/junos-vmhost-install-ptx-x86-64-15.1F5-
S2.8.tgz
Initializing...
  Verified os-libs-10-x86-64-20160616 signed by PackageProductionEc_2016
  Mounting os-libs-10-x86-64-20160616.329709_builder_stable_10
  ....
  Transfer Done
  Transfer /packages/db/pkginst.13874/junos-vmhost-install*.tgz
  Transfer Done
  Starting upgrade ...
  Preparing for upgrade...
  /tmp/pkg-0mc/unpack/install/
  ...
  ...
  Cmos Write successfull for Boot_retry
```

```
... upgrade complete.
A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY.
Use the 'request vmhost reboot' command to reboot the system
```

VM Host Rollback

You can revert to the software version that was loaded at the last successful `request vmhost software add` operation. You can roll back to the previous set of software packages, including the host OS packages, by using the `request vmhost software rollback` command.

The following example illustrates the software rollback operation. The Routing Engine that has booted from the primary disk by using the `set p` had booted using the `set b` before the upgrade.

```
user@host> show vmhost version
Current root details,          Device sda, Label: jrootp_P, Partition: sda3
Current boot disk: Primary
    Current root set: p
    EFI      Version: NGRE_v00.53.00.01
    Primary Disk, Upgrade Time: Wed Feb 24 17:51:53 UTC 2016
    Version: set p
    VMHost Version: 2.951
    VMHost Root: vmhost-x86_64-15.1I20160210_2212_builder
    VMHost Core: vmhost-core-x86_64-15.1I20160210_2212_builder
    kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
    Junos Disk: junos-install-x86-64-15.1F5.5
    Version: set b
    VMHost Version: 2.953
    VMHost Root: vmhost-x86_64-15.1F520160222_1052_builder
    VMHost Core: vmhost-core-x86_64-15.1F520160222_1052_builder
    kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
    Junos Disk: junos-install-x86-64-15.1F5.6
```

```
user@host> request vmhost software rollback
Current root details,          Device sda, Label: jrootp_P, Partition: sda3
    Finding alternate root for rollback
    Rollback to software on jrootb_P ...
    sh /etc/install/mk-mtre-rollback.sh jrootb_P b
    Mounting device in preparation for rollback...
```

```

Updating boot partition for rollback...
Rollback complete, please reboot the node for it to take effect.
Cmos Write successfull
Cmos Write successfull for Boot_retry
Cmos Write successfull for Boot_retry

```

```

user@host> show vmhost version
Current root details,          Device sda, Label: jrootp_P, Partition: sda3
    Current boot disk: Primary
    Current root set: p
    UEFI      Version: NGRE_v00.53.00.01
    Primary Disk, Upgrade Time: Wed Feb 24 17:51:53 UTC 2016
    Pending reboot.
    Version: set p
    VMHost Version: 2.951
    VMHost Root: vmhost-x86_64-15.1I20160210_2212_builder
    VMHost Core: vmhost-core-x86_64-15.1I20160210_2212_builder
    kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
    Junos Disk: junos-install-x86-64-15.1F5.5
    Version: set b
    VMHost Version: 2.953
    VMHost Root: vmhost-x86_64-15.1F520160222_1052_builder
    VMHost Core: vmhost-core-x86_64-15.1F520160222_1052_builder
    kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
    Junos Disk: junos-install-x86-64-15.1F5.6

```

```

user@host> request vmhost reboot
Reboot the vmhost ? [yes,no] (no) yes
    warning: Rebooting re1
    Initiating vmhost reboot... ok
    Initiating Junos shutdown... shutdown: [pid 9733]
    Shutdown NOW!
    ok
    Junos shutdown is in progress...
    *** FINAL System shutdown message ***

```

System going down IMMEDIATELY

```
user@host> show vmhost version
Current root details,      Device sda, Label: jrootb_P, Partition: sda4
  Current boot disk: Primary
  Current root set: b
  UEFI      Version: NGRE_v00.53.00.01
  Primary Disk, Upgrade Time: Wed Feb 24 17:51:53 UTC 2016
  Version: set p
  VMHost Version: 2.951
  VMHost Root: vmhost-x86_64-15.1I20160210_2212_builder
  VMHost Core: vmhost-core-x86_64-15.1I20160210_2212_builder
  kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
  Junos Disk: junos-install-x86-64-15.1F5.5
  Version: set b
  VMHost Version: 2.953
  VMHost Root: vmhost-x86_64-15.1F520160222_1052_builder
  VMHost Core: vmhost-core-x86_64-15.1F520160222_1052_builder
  kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
  Junos Disk: junos-install-x86-64-15.1F5.6
```

VM Host Snapshot

The snapshot feature enables you to create copies of the currently running and active file system partitions on a device.

On the device, you can back up the snapshot of the host OS image along with the Junos OS image. You can use the request `vmhost snapshot` command to create a VM host recovery snapshot on the backup disk.

Disk Recovery Using the VM Host Snapshot

If the state of the primary disk (disk1) is good and the backup disk (disk2) has to be recovered then use the request `vmhost snapshot` command to recover the backup disk assuming the Routing Engine is booted from the primary disk. If the state of the secondary disk is not known or the file systems in disk are not in a consistent state, then include partition option in the command i.e. request `vmhost snapshot partition`.

If the state of the backup disk (disk2) is good and the primary disk (disk1) has to be recovered then use the request `vmhost snapshot recovery` command to recover the primary disk assuming the Routing Engine is

booted from the backup disk. If the state of the primary disk is not known or the partition tables are in bad condition, then include partition option in the command i.e. `request vmhost snapshot recovery partition`.

To boot from desired disk, you can execute `request vmhost reboot { disk1, disk2}` command.

RELATED DOCUMENTATION

[Salient Features of the Routing Engines with VM Host Support | 372](#)

request vmhost software add

request vmhost software rollback

request vmhost snapshot

show vmhost snapshot

Copy the VM Host Installation Package to the PXE Boot Server

IN THIS CHAPTER

- [Copying VM Host Installation Package to the PXE Boot Server | 392](#)

Copying VM Host Installation Package to the PXE Boot Server

You can install the host OS, tools, and the Junos virtual machines (VMs) on the devices with RE-MX-X6, RE-MX-X8, RE-PTX-X8, and RE-QFX10002-60C, and RE-PTX10002-60C Routing Engines by using the Preboot Execution Environment (PXE) boot method. This is one of the methods used for a fresh installation. A PXE boot prepares a client/server environment to boot devices by using a network interface that is independent of available data storage devices or installed operating systems. The image of the operating system is stored on a TFTP server.

To copy the installation packages to the PXE boot server:

1. Copy the downloaded installation media to the **/var/tmp** directory in the PXE boot server.

```
scp /volume/build/junos/15.1/release/15.1F3.9/ship/junos-vmhost-install-net-  
x86-64-15.1F3.9.tgz user@host:/var/tmp/
```

2. Log in to the PXE boot server and verify the installation file.

```
user@host> ls -lh junos-vmhost-install-net-x86-64-15.1F3.9.tgz  
-rw-r--r-- 1 root root 1.8G Oct 24 00:42 junos-vmhost-install-net-x86-64-15.1F3.9.tgz
```

3. Extract the **junos-vmhost-install-net** TAR file.

```
user@host> tar xvzf junos-vmhost-install-net-x86-64-15.1F3.9.tgz -C /var/tmp  
contents/  
contents/junos-vmhost-install.tgz
```



```

contents/vmhost-install-net-x86_64-15.1I20151019_1021_builder.tgz
manifest
manifest.certs
manifest.ecerts
manifest.esig
manifest.sig
package.xml

```

4. Remove the previously installed files, if any, from the **/tftpboot** directory.

```

user@host> rm -f /tftpboot/{vmhost-version.sh,bootpxe64.efi,vmhost-
version,grub.cfg,initramfs,vmlinuz}
user@host>ls -lh /tftpboot//
total 45M

-rw-r--r-- 1 root root 690K Sep  8 13:22 bootpxe.efi
-rw-rw-r-- 1 930  930  45M Oct 20 01:51 vmhost-install-net-
x86_64-15.1I20151019_1021_builder.tgz

```

5. Extract the network installation package.

```

user@host> tar xvzf /var/tmp/contents/vmhost-install-net-
x86_64-15.1I20151019_1021_builder.tgz -C /tftpboot/
./
./vmhost-version.sh
./bootpxe64.efi
./vmhost-version
./grub.cfg
..
...
-rw-rw-r-- 1 930  930  45M Oct 20 01:51 vmhost-install-net-
x86_64-15.1I20151019_1021_builder.tgz
-rw-rw-r-- 1 930  930    6 Oct 20 01:51 vmhost-version
-rwxrwxr-x 1 930  930  416 Oct 20 01:51 vmhost-version.sh
-rw-r--r-- 1 930  930 6.9M Oct 20 01:51 vmlinuz

```

6. Rename or delete the previously installed root file system/scripts from the **/var/install** directory. Create a new **/var/install** directory.

```
user@host>mv /var/install /var/install_old
user@host>mkdir /var/install
```

7. Extract the installation package.

```
user@host>tar xvzf /var/tmp/contents/junos-vmhost-install.tgz -C /var/install
./
./vmhost-pkgs-version
./vm/
./vm/note
./vm/grub.cfg.ngre
./vm/vsmartd-1.0-0.x86_64.rpm
./vm/re_fpga-1.0-0.x86_64.rpm
./vm/veccd-1.0-0.x86_64.rpm
./vmhost-version.sh
./vmhost/
./vmhost/vmhost-x86_64-15.1I20151019_1021_builder.img.gz
...
...
./junos/junos-mtre-upgrade.sh
./vmhost-core-x86_64-15.1I20151019_1021_builder.tgz
./junos/
./junos/junos-install-x86-64-15.1F3.9.img.gz
```

8. Set permissions for the files in the **/var/install** and **/tftpboot** directories.

```
user@host> chown root:root /tftpboot/*
user@host> chmod a+rwX /tftpboot/*
user@host> chown -R root:root /var/install
user@host> chmod -R a+rwX /var/install
```

9. Exit the PXE boot server.

```
user@host> exit
```

RELATED DOCUMENTATION

[Installing, Upgrading, Backing Up, and Recovery of VM Host | 385](#)

[Creating an Emergency Boot Device for Routing Engines with VM Host Support | 416](#)

Upgrade NFX Devices

IN THIS CHAPTER

- Upgrade the NFX250 Software to NFX250 NextGen Software | 396
- Upgrading the Junos OS on NFX Devices | 399
- Upgrade Dual-Disk Partitions on NFX250 NextGen and NFX350 Devices | 404
- Downgrade Instructions for NFX Series Devices Running Junos OS Release 23.1R1 | 415

Upgrade the NFX250 Software to NFX250 NextGen Software

IN THIS SECTION

- NFX250 NextGen Software Upgrade Overview | 396
- Prerequisites | 397
- Upgrade to NFX250 NextGen Software Architecture | 399

NFX250 NextGen Software Upgrade Overview

Starting in Junos OS Release 19.1R1, the NFX250 devices support the NFX250 NextGen software architecture. This is a re-optimized architecture that enables you to use JCP as the single point of management to manage all the NFX250 components. For more information about the NFX250 NextGen architecture, see [NFX250 NextGen Overview](#).



NOTE: For documentation purposes, NFX250 devices that use the reoptimized architecture are referred to as NFX250 NextGen devices.

You can upgrade the software using a USB or through a CLI. This topic provides information about prerequisites and the procedure to upgrade through a CLI from NFX250 software architecture to NFX250 NextGen software architecture.



NOTE: The upgrade procedure using a USB remains the same for all NFX Series devices.

Prerequisites

To upgrade an NFX250 device, you must meet the following prerequisites:

Device-specific prerequisites

- An NFX250 device with BIOS => CBDE_SFP_00.21_01.01

To verify the BIOS version:

```
root@jdm> request execute-command "jhost dmidecode -t bios"
```

For the BIOS information, see the BIOS Information section in the command output message.

If the BIOS version is not CBDE_SFP_00.21_01.01, you can upgrade the BIOS:

1. Download the BIOS from [Downloads](#) page.
2. Copy and save the BIOS image to the **/var/third-party** directory.
3. From the JDM CLI, access the hypervisor:

```
root@jdm> ssh hypervisor
```

4. Upgrade the BIOS:

```
root@host:~# rpm -ivh /var/third-party/firmware/BIOS RPM package name
```

The system generates the following output:

```
Preparing...          ##### [100%]
1:nfx-2-secure-bios   ##### [100%]
A reboot is required to install the secure BIOS
Please reboot the system to complete the install
```

5. Reboot the device to load new BIOS.

a. Exit from hypervisor shell:

```
root@local-node:~# exit
logout
Connection to hypervisor closed.
{master:0}
root@JDM>
```

b. Reboot the device from JDM CLI.

```
{master:0}
root@porter-p2a-sys1> request system reboot
Reboot the system ? [yes,no] (no) yes
```

- An NFX250 NextGen configuration file with minimal or necessary configurations is required for remote management access to the device after migrating to NFX250 NextGen. This file is an input data for the `request system software add clean-install package-name` command.

Release-specific prerequisites

The NFX250 software must be compatible with the following releases:

- NFX250 software running Junos OS Release 18.4R2 or later to accept the configuration by using the command:

```
user@host> request system software add clean-install package-name
```



CAUTION: The `clean-install` command removes all contents on the hard disk. To avoid data loss, copy all important files, configuration files (JDM, JCP, vSRX Virtual Firewall, and third-party VNFs), log files, and VNF disk or image file, and save them in a secure location before you upgrade the device.

- Releases prior to 18.4R2 must be upgraded to 18.4R2 or later.



CAUTION: The NFX250 device will crash if you upgrade the NFX250 software image running Junos OS Release prior to 18.4R2 to a release that supports NFX250 NextGen software image.

The NFX250 NextGen configuration must be compatible with the NFX250 NextGen software version. The configuration command syntax is not validated.



NOTE: The NFX250 software architecture and NFX250 NextGen software architecture are different and the configurations are different for both the software.

Upgrade to NFX250 NextGen Software Architecture

Before you upgrade the NFX device:

- Create backup of the configuration files (JDM, JCP, vSRX Virtual Firewall, and third-party VNFs), log files, VNF disk or image file, and other important files stored on the device.
- Check the prerequisites.

To upgrade the NFX250 software architecture to NFX250 NextGen software architecture:

1. Copy the configuration files that are required for in-band and out-of-band management and save it in the **/var/third-party** folder. The configuration file should be of the same format as the file format obtained by running the `show configuration` CLI command.
2. Copy the NFX250 NextGen software image and save it in the **/var/third-party/images** folder.
3. Initiate the software upgrade by using the following command:

```
root@jdm> request system software add clean-install reboot /var/third-party/images/jinstall-  
image.tgz upgrade-with-config /var/third-party/config-file
```

The device is formatted and the NFX250 NextGen software image is installed. The device loads the configurations and boots up the NFX250 Nextgen software image. You can access the device remotely through the in-band and out-of-band management.

4. The device is now ready for additional configurations and third-party VNF onboarding.

Upgrading the Junos OS on NFX Devices

To upgrade the Junos OS version on NFX150, NFX250 NextGen, and NFX350 devices:

1. Download the software package from the [Downloads](#) page to the `/var/public` directory on the NFX device.
2. Verify the Junos OS version that is currently installed on the device. The following sample output for NFX250 NextGen shows that Junos OS Release 22.3R2.12 is installed on the device.

```

user@host> show version
root@host> show version
Hostname: host
Model: nfx250
Junos: 22.3R2.12
JUNOS OS Kernel 64-bit [20221212.98a33a0_builder_stable_12_223]
JUNOS OS libs [20221212.98a33a0_builder_stable_12_223]
JUNOS OS runtime [20221212.98a33a0_builder_stable_12_223]
JUNOS OS time zone information [20221212.98a33a0_builder_stable_12_223]
JUNOS network stack and utilities [20230223.221505_builder_junos_223_r2]
JUNOS libs [20230223.221505_builder_junos_223_r2]
JUNOS OS libs compat32 [20221212.98a33a0_builder_stable_12_223]
JUNOS OS 32-bit compatibility [20221212.98a33a0_builder_stable_12_223]
JUNOS libs compat32 [20230223.221505_builder_junos_223_r2]
JUNOS runtime [20230223.221505_builder_junos_223_r2]
JUNOS Packet Forwarding Engine Simulation Package [20230223.221505_builder_junos_223_r2]
JUNOS sflow mx [20230223.221505_builder_junos_223_r2]
JUNOS py extensions [20230223.221505_builder_junos_223_r2]
JUNOS py base [20230223.221505_builder_junos_223_r2]
JUNOS OS vmguest [20221212.98a33a0_builder_stable_12_223]
JUNOS OS package [20230213.192558_builder_stable_12]
JUNOS OS crypto [20221212.98a33a0_builder_stable_12_223]
JUNOS OS boot-ve files [20221212.98a33a0_builder_stable_12_223]
JUNOS na telemetry [22.3R2.12]
JUNOS Wireless WAN Module [20230223.221505_builder_junos_223_r2]

```

You can use the `show vmhost version detail` command to view the Junos OS version that is installed on the disk partitions. Note that the NFX150 and NFX250 NextGen have a single disk with two partitions whereas the NFX350 has dual disks that provide four partitions.

Here's a sample output for an NFX250 NextGen device. You'll notice that the primary partition is the active partition.

```

user@host> show vmhost version detail
Partition set      : primary
Software version   : 22.3R2.12

```



```
Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
Host kernel version : #1 SMP Sun Dec 4 22:30:10 PST 2022
```

```
Partition set      : primary
Software version   : 22.3R2.12
Installed/Upgraded at : Tue Feb 28 10:12:50 UTC 2023
Status             : Boot success

Partition set      : alternate
Software version   : 22.3R2.12
Installed/Upgraded at : Tue Feb 28 10:17:48 UTC 2023
Status             : Factory installation setup complete, ready for boot
```

3. Upgrade the Junos OS by using the request vmhost software command.

```
user@host> request vmhost software add /var/public/jinstall-host-nfx-3-x86-64-22.4R2.1-secure-
signed.tgz
Verified jinstall-host-nfx-3-x86-64-22.4R2.1-secure-signed signed by
PackageProductionECP256_2023 method ECDSA256+SHA256
Pushing Junos image package to the host...
File already present in Host. Skipping pushing the image
Mounting alternate partitions to stage upgrade operation
Installing /var/tmp/preinstall/install-media-nfx-3-junos-22.4R2.1-secure.tgz
Extracting the package ...
Validate linux image...
upgrade_platform: -----
upgrade_platform: Parameters passed:
upgrade_platform: silent=0
upgrade_platform: package=/var/tmp/tmp.27UZct4newjunos_cli_upg/jinstall-nfx-3-junos-22.4R2.1-
secure-linux.tgz
upgrade_platform: clean install=0
upgrade_platform: Need reboot after staging=1
upgrade_platform: -----
upgrade_platform:
upgrade_platform: Checking input /var/tmp/tmp.27UZct4newjunos_cli_upg/jinstall-nfx-3-
junos-22.4R2.1-secure-linux.tgz ...
upgrade_platform: Input package /var/tmp/tmp.27UZct4newjunos_cli_upg/jinstall-nfx-3-
junos-22.4R2.1-secure-linux.tgz is valid.
Secure Boot is enforced.
ALLOW:usr/secureboot/grub/BOOTX64.EFI
ALLOW:boot/bzImage-intel-x86-64.bin
ALLOW:boot/initramfs.cpio.gz
```

```

Setting up Junos host applications for installation ...
Installing Host OS ...
upgrade_platform: -----
upgrade_platform: Parameters passed:
upgrade_platform: silent=0
upgrade_platform: package=/var/tmp/tmp.27UZct4newjunos_cli_upg/jinstall-nfx-3-junos-22.4R2.1-secure-linux.tgz
upgrade_platform: clean install=0
upgrade_platform: Need reboot after staging=0
upgrade_platform: -----
upgrade_platform:
upgrade_platform: Checking input /var/tmp/tmp.27UZct4newjunos_cli_upg/jinstall-nfx-3-junos-22.4R2.1-secure-linux.tgz ...
upgrade_platform: Input package /var/tmp/tmp.27UZct4newjunos_cli_upg/jinstall-nfx-3-junos-22.4R2.1-secure-linux.tgz is valid.
Secure Boot is enforced.
ALLOW:usr/secureboot/grub/BOOTX64.EFI
ALLOW:boot/bzImage-intel-x86-64.bin
ALLOW:boot/initramfs.cpio.gz
upgrade_platform: Backing up boot assets..
upgrade_platform: Staging the upgrade package - /var/tmp/tmp.27UZct4newjunos_cli_upg/jinstall-nfx-3-junos-22.4R2.1-secure-linux.tgz..
upgrade_platform: Checksum verified and OK...
Setting up boot environment ...
Setting up boot environment completed
upgrade_platform: Staging of /var/tmp/tmp.27UZct4newjunos_cli_upg/jinstall-nfx-3-junos-22.4R2.1-secure-linux.tgz completed
upgrade_platform: System needs *REBOOT* to complete the upgrade

Host OS upgrade staged. Reboot the system to complete installation!

```

If you run the `show vmhost version` command now, you notice that the software version still shows 22.3R2.12. Note that the status of the alternate partition indicates that the new Junos OS version is staged on the alternate partition.

```

root@host> show vmhost version detail | no-more
Partition set      : primary
Software version   : 22.3R2.12
                   Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Sun Dec 4 22:30:10 PST 2022
Reboot is pending for a software upgrade

```

```

Partition set      : primary
Software version   : 22.3R2.12
Installed/Upgraded at : Tue Feb 28 10:12:50 UTC 2023
Status             : Boot success

Partition set      : alternate
Software version   : 22.3R2.12
Installed/Upgraded at : Tue Feb 28 10:17:48 UTC 2023
Status             : Software upgrade staged, boot to partition-set pending

```

4. Reboot the device to load the new version of Junos OS on the device.

```

root@host> request vmhost reboot
Reboot the vmhost ? [yes,no] (no) yes

Initiating vmhost reboot...

root@host> show version
Hostname: host
Model: nfx250_att_ls1_10_t
Junos: 22.4R2.1
JUNOS OS Kernel 64-bit [20230213.5295c32_builder_stable_12_224]
JUNOS OS libs [20230213.5295c32_builder_stable_12_224]
JUNOS OS runtime [20230213.5295c32_builder_stable_12_224]
JUNOS OS time zone information [20230213.5295c32_builder_stable_12_224]
JUNOS network stack and utilities [20230301.213842_builder_junos_224_r2]
JUNOS libs [20230301.213842_builder_junos_224_r2]
JUNOS OS libs compat32 [20230213.5295c32_builder_stable_12_224]
JUNOS OS 32-bit compatibility [20230213.5295c32_builder_stable_12_224]
JUNOS libs compat32 [20230301.213842_builder_junos_224_r2]
JUNOS runtime [20230301.213842_builder_junos_224_r2]
JUNOS Packet Forwarding Engine Simulation Package [20230301.213842_builder_junos_224_r2]
JUNOS sflow mx [20230301.213842_builder_junos_224_r2]
JUNOS py extensions [20230301.213842_builder_junos_224_r2]
JUNOS py base [20230301.213842_builder_junos_224_r2]
JUNOS OS vmguest [20230213.5295c32_builder_stable_12_224]
JUNOS OS package [20230213.192558_builder_stable_12]
JUNOS OS crypto [20230213.5295c32_builder_stable_12_224]
JUNOS OS boot-ve files [20230213.5295c32_builder_stable_12_224]
JUNOS na telemetry [22.4R2.1]

```

5. After the device reboots, verify the status of the disk partitions. The following output shows that the image on the alternate partition is upgraded. The alternate partition is the active partition now.

```

root@host> show vmhost version detail
Partition set      : alternate
Software version   : 22.4R2.1
                   Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Mon Feb 27 04:48:57 PST 2023

Partition set      : primary
Software version   : 22.3R2.12
Installed/Upgraded at : Tue Feb 28 10:12:50 UTC 2023
Status             : Boot success

Partition set      : alternate
Software version   : 22.4R2.1
Installed/Upgraded at : Tue Mar  7 16:05:24 UTC 2023
Status             : Boot success

```

To upgrade the release to the same version on all the disk partitions, see [Upgrading Dual-Disk Partitions on NFX250 NextGen and NFX350 Devices](#).

Upgrade Dual-Disk Partitions on NFX250 NextGen and NFX350 Devices

SUMMARY

IN THIS SECTION

- Upgrading Disk Partitions Using the request vmhost software add *package-name* Command on an NFX350 Device | **405**
- Upgrading Disk Partitions Using the request vmhost software add *package-name* Command on an NFX250 NextGen Device | **411**
- Upgrading Disk Partitions Using the clean-install Command | **412**

- [Upgrading Disk Partitions Using the request system zeroize Command | 414](#)
- [Upgrading Disk Partitions Using a USB | 415](#)

You can upgrade the disk partitions on NFX250 NextGen and NFX350 devices by using:

- The request vmhost software add *package-name* command.
- The request system software add clean-install *package-name* command.
- The request system zeroize command.
- The USB image installation method.

Upgrading Disk Partitions Using the request vmhost software add *package-name* Command on an NFX350 Device

When you upgrade the disks using this method, the device retains all the configuration and log information.

To upgrade the disk partitions using the request vmhost software add *package-name* command:

1. Verify the initial status of the disks.

```
user@host> show vmhost version detail
Partition set      : primary
Software version   : 21.4I-20220531.0.1918
                   Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Mon Nov 29 03:34:19 PST 2021

Partition set      : primary
Software version   : 21.4I-20220531.0.1918
Installed/Upgraded at : Thu Jun  2 05:09:21 PDT 2022
Status             : Boot success

Partition set      : alternate
Software version   : 22.3I20220517_0906
Installed/Upgraded at : Wed May 18 21:45:46 PDT 2022
Status             : Boot success

Partition set      : second primary
```

```

Software version      : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 00:50:58 PST 2022
Status                : Factory installation setup complete, ready for boot

Partition set        : second alternate
Software version      : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 00:51:01 PST 2022
Status                : Factory installation setup complete, ready for boot

```

2. Upgrade the device with the Junos OS image version that supports the dual disk upgrade.

When prompted for a reboot, type Yes.

```

user@host> request vmhost software add /var/public/jinstall-host-nfx-3-x86-64-21.4R2.8-
secure-signed.tgz

```

3. After the device reboots, verify the status of the disk partitions. The output shows that the image on the alternate partition of disk 1 is upgraded. The alternate partition is the active partition now.

```

user@host> show vmhost version detail
Partition set      : alternate
Software version   : 21.4R2.8
                   Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Tue Feb 22 01:50:05 PST 2022

Partition set      : primary
Software version    : 21.4I-20220531.0.1918
Installed/Upgraded at : Thu Jun  2 05:09:21 PDT 2022
Status              : Boot success

Partition set      : alternate
Software version    : 21.4R2.8
Installed/Upgraded at : Sun Jul  3 22:38:50 PDT 2022
Status              : Boot success

Partition set      : second primary
Software version    : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 00:50:58 PST 2022
Status              : Factory installation setup complete, ready for boot

Partition set      : second alternate

```

```

Software version      : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 00:51:01 PST 2022
Status                : Factory installation setup complete, ready for boot

```

4. Upgrade the device again to upgrade the primary partition of disk 1.

When prompted for a reboot, type Yes.

```

user@host> request vmhost software add /var/public/jinstall-host-nfx-3-x86-64-21.4R2.8-
secure-signed.tgz

```

5. After the device reboots, verify the status of the disk partitions. The output shows that the image on the primary partition of disk 1 is upgraded. Both the primary and alternate partitions on disk 1 are running the same image. Note that the primary partition is the active partition now.

```

user@host> show vmhost version detail
Partition set      : primary
Software version   : 21.4R2.8
                   Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Tue Feb 22 01:50:05 PST 2022

Partition set      : primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 00:02:33 PDT 2022
Status             : Boot success

Partition set      : alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Sun Jul  3 22:38:50 PDT 2022
Status             : Boot success

Partition set      : second primary
Software version   : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 00:50:58 PST 2022
Status             : Factory installation setup complete, ready for boot

Partition set      : second alternate
Software version   : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 00:51:01 PST 2022

```

```
Status : Factory installation setup complete, ready for boot
```

6. To upgrade the secondary disk, switch to the primary partition of disk 2.



NOTE: Before you switch from disk 1 to disk 2, ensure that the required basic configuration (such as management connectivity) is available on disk 2.

```
user@host> request vmhost reboot disk2 primary
```

7. After the device reboots, verify the status of the disk partitions.

```
user@host> show vmhost version detail
Partition set      : second primary
Software version   : 20.4R3.8
                   Host kernel release : 4.1.27-rt30-WR8.0.0.32_ovp
                   Host kernel version  : #1 SMP Thu Jul 8 23:25:47 PDT 2021

Partition set      : primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 07:02:33 UTC 2022
Status             : Boot success

Partition set      : alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 07:51:48 UTC 2022
Status             : Boot success

Partition set      : second primary
Software version   : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 08:50:58 UTC 2022
Status             : Boot success

Partition set      : second alternate
Software version   : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 08:51:01 UTC 2022
Status             : Factory installation setup complete, ready for boot
```


8. Upgrade the device with the Junos OS image version that supports the dual disk upgrade.

```
user@host> request vmhost software add /var/public/jinstall-host-nfx-3-x86-64-21.4R2.8-secure-signed.tgz
```

9. Verify the status of the disk partitions after the upgrade. The output shows that the image on the alternate partition of disk 2 is upgraded. The alternate partition of disk 2 is the active partition now.

```
user@host> show vmhost version detail
Partition set      : second alternate
Software version   : 21.4R2.8
                   Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Tue Feb 22 01:50:05 PST 2022

Partition set      : primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 07:02:33 UTC 2022
Status             : Boot success

Partition set      : alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 07:51:48 UTC 2022
Status             : Boot success

Partition set      : second primary
Software version   : 20.4R3.8
Installed/Upgraded at : Fri Jan  7 08:50:58 UTC 2022
Status             : Boot success

Partition set      : second alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 10:21:55 UTC 2022
Status             : Boot success
```

10. Upgrade the primary partition on disk 2.

```
user@host> request vmhost software add /var/public/jinstall-host-nfx-3-x86-64-21.4R2.8-secure-signed.tgz
```

11. Verify the status of the disk partitions after the upgrade. The output shows that the image on the primary partition of disk 2 is upgraded. Both the primary and alternate partitions on disk 2 are running the same image. The primary partition on disk 2 is the active partition now.

```
user@host> show vmhost version detail

Partition set      : second primary
Software version   : 21.4R2.8
                    Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                    Host kernel version  : #1 SMP Tue Feb 22 01:50:05 PST 2022

Partition set      : primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 07:02:33 UTC 2022
Status             : Boot success

Partition set      : alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 07:51:48 UTC 2022
Status             : Boot success

Partition set      : second primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 11:00:05 UTC 2022
Status             : Boot success

Partition set      : second alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 10:21:55 UTC 2022
Status             : Boot success
```

Upgrading Disk Partitions Using the request vmhost software add *package-name* Command on an NFX250 NextGen Device

When you upgrade the disks using this method, the device retains all the configuration and log information.

To upgrade the disk partitions using the request vmhost software add *package-name* command:

1. Upgrade the device with the Junos OS image version which supports the dual disk upgrade using the request vmhost software add *package-name* command.

When prompted for a reboot, type Yes.

```
user@host> request vmhost software add /var/public/jinstall-host-nfx-3-
x86-64-22.3I-20220428_dev_common.0.0158-secure-signed.tgz
```

2. Verify the status of the disks. The output shows that the image on the alternate partition is upgraded. The alternate partition is the active partition now.

```
user@host> show vmhost version detail
Partition set      : alternate
Software version   : 22.3I-20220428_dev_common.0.0158
                   Host kernel release  : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Sun Apr 24 23:33:52 PDT 2022

Partition set      : primary
Software version   : 21.1R3.11
Installed/Upgraded at : Wed Dec 22 02:13:33 PST 2021
Status             : Boot success

Partition set      : alternate
Software version   : 22.3I-20220428_dev_common.0.0158
Installed/Upgraded at : Thu Apr 28 20:47:21 PDT 2022
Status             : Boot success
```

3. Upgrade the device again to upgrade the primary partition.

When prompted for a reboot, type Yes.

```
user@host> request vmhost software add /var/public/jinstall-host-nfx-3-
x86-64-22.3I-20220428_dev_common.0.0158-secure-signed.tgz
```

4. After the reboot, verify the status of the disk partitions. The output shows that the image on the primary partition is upgraded. The primary and alternate partitions are now running the same image, Note that the primary partition is the active partition now.

```

user@host> show vmhost version detail
Partition set      : primary
Software version   : 22.3I-20220428_dev_common.0.0158
                   Host kernel release  : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Sun Apr 24 23:33:52 PDT 2022

Partition set      : primary
Software version   : 22.3I-20220428_dev_common.0.0158
Installed/Upgraded at : Wed Jul 13 23:56:23 PDT 2022
Status             : Boot success

Partition set      : alternate
Software version   : 22.3I-20220428_dev_common.0.0158
Installed/Upgraded at : Thu Apr 28 20:47:21 PDT 2022
Status             : Boot success

```

Upgrading Disk Partitions Using the clean-install Command

This upgrade method resets all the log and configuration information, and loads the specified image on all the disk partitions.

To upgrade the disk partitions using the `clean-install` command:

1. Verify the initial status of the disks.

```

user@host> show vmhost version detail
Partition set      : primary
Software version   : 21.4R2.8
                   Host kernel release  : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Tue Feb 22 01:50:05 PST 2022

Partition set      : primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul 4 00:02:33 PDT 2022
Status             : Boot success

```

```

Partition set      : alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 00:51:48 PDT 2022
Status             : Boot success

Partition set      : second primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 04:00:05 PDT 2022
Status             : Boot success

Partition set      : second alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 03:21:55 PDT 2022
Status             : Boot success

```

2. Upgrade the device by using the clean-install command.

```

user@host> request vmhost software add /var/public/jinstall-host-nfx-3-
x86-64-21.4I-20220531.0.1918-secure-signed.tgz clean-install

```

3. Verify the disk details after the upgrade. The output shows that the image on all the disk partitions is upgraded.

```

user@host> show vmhost version detail
Partition set      : primary
Software version   : 21.4I-20220531.0.1918
                   Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Mon Nov 29 03:34:19 PST 2021

Partition set      : primary
Software version   : 21.4I-20220531.0.1918
Installed/Upgraded at : Tue Jul  5 04:10:39 UTC 2022
Status             : Boot success

Partition set      : alternate
Software version   : 21.4I-20220531.0.1918
Installed/Upgraded at : Tue Jul  5 04:16:37 UTC 2022
Status             : Factory installation setup complete, ready for boot

Partition set      : second primary
Software version   : 21.4I-20220531.0.1918
Installed/Upgraded at : Tue Jul  5 04:17:27 UTC 2022

```

```

Status                : Factory installation setup complete, ready for boot

Partition set         : second alternate
Software version      : 21.4I-20220531.0.1918
Installed/Upgraded at : Tue Jul  5 04:17:24 UTC 2022
Status                : Factory installation setup complete, ready for boot

```

Upgrading Disk Partitions Using the request system zeroize Command

This upgrade method resets all the log and configuration information, and loads the image running on the current active partition on all the other disk partitions.

To upgrade the disk partitions:

1. Remove all configuration information on the Routing Engines and reset all key values.

```
user@host> request system zeroize
```

2. Verify the disk details. The output shows that the image on all the disk partitions is upgraded.

```

user@host> show vmhost version detail
Partition set      : primary
Software version   : 21.4R2.8
                   Host kernel release : 4.1.27-rt30-WR8.0.0.34_ovp
                   Host kernel version  : #1 SMP Tue Feb 22 01:50:05 PST 2022

Partition set      : primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 00:02:33 PDT 2022
Status             : Boot success

Partition set      : alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 00:51:48 PDT 2022
Status             : Boot success

Partition set      : second primary
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 04:00:05 PDT 2022
Status             : Boot success

```

```

Partition set      : second alternate
Software version   : 21.4R2.8
Installed/Upgraded at : Mon Jul  4 03:21:55 PDT 2022
Status            : Boot success

```

Upgrading Disk Partitions Using a USB

For information about upgrading the disk partitions using a USB, see [KB31834](#).

Downgrade Instructions for NFX Series Devices Running Junos OS Release 23.1R1

On the NFX150, NFX250 NextGen, and NFX350 devices, you cannot downgrade Junos OS Release 23.1R1 directly to certain releases (listed in the **Target Release** column in [Table 22 on page 415](#)). As a workaround, you can perform downgrade as a two-step activity, as described below:

1. Downgrade the Junos OS Release 23.1R1 software to the corresponding intermediate release.
2. Downgrade the software from the intermediate release to the target release.

[Table 22 on page 415](#) lists the target releases and the corresponding intermediate releases.

Table 22: Release Compatibility for Downgrading Junos OS 23.1R1 on NFX Series Devices

Target Release	Intermediate Release
Any 22.4x release earlier than 22.4R2	22.4R2
Any 22.3x release earlier than 22.3R2.	22.3R2
<ul style="list-style-type: none"> Any 22.2x release earlier than 22.2R3. Any 22.1x release or earlier releases. 	22.2R3

Create an Emergency Boot Device for Routing Engines with VM Host Support

IN THIS CHAPTER

- [Creating an Emergency Boot Device for Routing Engines with VM Host Support | 416](#)

Creating an Emergency Boot Device for Routing Engines with VM Host Support

If Junos OS on your device is damaged during loading in a way that prevents it from loading completely, you can use the emergency boot device to revive the device. The emergency boot device repartitions the primary disk and reloads a fresh installation of Junos OS. For RE-MX-X6, RE-MX-X8, RE-PTX-X8, and RCBPTX Routing Engines, you can use a USB storage device with at least 8 GB of free space to create an emergency boot device.

To create an emergency boot device on a device with RE-MX-X6, RE-MX-X8, RE-PTX-X8, RCBPTX, RE-QFX10002-60C, and RE-PTX10002-60C Routing Engines:

1. Copy the installation media into the device's **/var/tmp** directory.
2. Insert the USB storage device into the device's USB port.
3. In the UNIX shell, navigate to the **/var/tmp** directory:

```
start shell
cd /var/tmp
```

4. Log in as su:

Super User (su) is one of the predefined login classes with preset permissions.

```
su [enter]
password: [enter SU password]
```


5. Gunzip the copied file.

For example, to convert `junos-vmhost-install-usb-mx-x86-64-15.1F6.8.img.gz` to `junos-vmhost-install-usb-mx-x86-64-15.1F6.8.img`, use the following command: `gunzip junos-vmhost-install-usb-mx-x86-64-15.1F6.8.img.gz`

6. Issue the following command:

```
dd if=/path/to/downloaded.img of=/dev/devicenode bs=4M
```

where:

- *devicenode*—Refers to the name of the removable media of the emergency boot device. For names of storage media, see ["Routing Engines and Storage Media Names \(ACX Series, M Series, MX Series, PTX Series, T Series, TX Matrix, TX Matrix Plus, and JCS 1200 Routers\)"](#) on page 32.
- *downloaded.img*—Refers to the installation media copied to the `/var/tmp` directory. For example, `junos-vmhost-install-usb-ptx-x86-64-15.1F6.8.img`.

The following code example can be used to create an emergency boot device by using a USB storage device:

```
dd if=/path/to/junos-vmhost-install-usb-mx-x86-64-15.1F6.8.img of=/dev/da0 bs=4M
```



NOTE: In the `dd` command, use `junos-vmhost-install-usb-mx-86` for RE-MX-X6 and RE-MX-X8 Routing Engines and `junos-vmhost-install-ptx-86` for RE-PTX-X8 Routing Engine respectively.

7. Log out as `su`:

```
exit
```

RELATED DOCUMENTATION

| [Boot Process for Routers with VM Host Support](#) | 382

Upgrade Firmware on Routing Engines with VM Host Support

IN THIS CHAPTER

- [Upgrading the SSD Firmware on Routing Engines with VM Host Support | 418](#)
- [Upgrading the i40e NVM Firmware on Routing Engines with VM Host Support | 421](#)

Upgrading the SSD Firmware on Routing Engines with VM Host Support

Starting in Junos OS Release 17.2R1, you can upgrade the solid-state drive (SSD) firmware on MX Series routers with the RE-S-X6-64G and RE-MX2K-X8-64G Routing Engines, on QFX10002-60C switches with the RE-QFX10002-60C Routing Engines, and PTX10002-60C routers with the RE-PTX10002-60C Routing Engines. A set of two SSDs, disk1 and disk2 , is available for normal functioning of the Routing Engine. This topic shows how to perform the upgrade.



NOTE: You must upgrade SSD firmware only under the direction of a Juniper Networks support representative.



NOTE: On QFX10002-60C switches, you can upgrade firmware only for the FPGA and BIOS, not the SSD.

Before you begin upgrading the firmware, check the current firmware version of the SSD.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 0	RE BIOS	0	0.45	0.53	OK
Routing Engine 0	RE FPGA	1	36.0.0	41.0	OK
Routing Engine 0	RE SSD1	4	12028	12029	OK

Routing Engine	0	RE SSD2	5	12028	12029	OK
Routing Engine	1		0	1.4		OK

If the value of Current version is less than the value of Available version, then you can use the following procedure for the SSD firmware upgrade.

To upgrade SSD firmware:

1. Copy the jfirmware package to the device.

If the file has been obtained from JTAC, use FTP or SCP to load the firmware file on the device. Save the file in the /var/tmp directory.

```
user@host> request system software add ftp://ftp.juniper.net/private/system/jfirmware-17.1R2-signed.tgz
```

2. Upgrade the SSD disk1 firmware.



NOTE: In releases before Junos OS Release 18.3R1, you must upgrade the SSD on a primary Routing Engine only. For upgrading firmware on the backup Routing Engine, switch primary role by using the following command and then log in to the backup Routing Engine, which is now the new primary Routing Engine..

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

Starting in Junos OS Release 18.3R1, you can upgrade the SSD firmware on the primary and backup Routing Engines.

To initiate the upgrade, use the following command:

```
user@host> request system firmware upgrade re ssd disk1
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine	0	RE SSD1	4	12028 12029	OK

Perform indicated firmware upgrade ? [yes,no] (no) yes

Firmware upgrade initiated, use "show system firmware" to monitor status.

Monitor the upgrade status by using the show system firmware command.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine	0	RE BIOS	0	0.45 0.53	OK

```

Routing Engine 0    RE FPGA  1      36.0.0  41.0    OK
Routing Engine 0    RE SSD1  4      12028   12029   OK
Routing Engine 0    RE SSD2  5      12028   12029   OK
Routing Engine 1           0      1.4           OK

```

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 0	RE BIOS	0	0.45	0.53	OK
Routing Engine 0	RE FPGA	1	36.0.0	41.0	OK
Routing Engine 0	RE SSD1	4	12029	12029	UPGRADED SUCCESSFULLY
Routing Engine 0	RE SSD2	5	12028	12029	OK
Routing Engine 1		0	1.4		OK

After a successful upgrade, confirm that the current version and available version of the SSD firmware are identical.

3. Upgrade SSD Disk2 firmware.

To initiate the upgrade, use the following command:

```
user@host> request system firmware upgrade re ssd disk2
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 0	RE SSD2	5	12028	12029	OK

Perform indicated firmware upgrade ? [yes,no] (no) yes

Firmware upgrade initiated, use "show system firmware" to monitor status.

Monitor the upgrade status by using the show system firmware command.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 0	RE BIOS	0	0.45	0.53	OK
Routing Engine 0	RE FPGA	1	36.0.0	41.0	OK
Routing Engine 0	RE SSD1	4	12028	12029	UPGRADED SUCCESSFULLY

Routing Engine	0	RE SSD2	5	12028	12029	PROGRAMMING
Routing Engine	1		0	1.4		OK

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine	0	RE BIOS	0	0.45	0.53 OK
Routing Engine	0	RE FPGA	1	36.0.0	41.0 OK
Routing Engine	0	RE SSD1	4	12029	12029 UPGRADED SUCCESSFULLY
Routing Engine	0	RE SSD2	5	12029	12029 UPGRADED SUCCESSFULLY
Routing Engine	1		0	1.4	OK

After a successful upgrade, confirm that the current version and available version of the SSD firmware are identical.

Change History Table

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

Release	Description
17.2R1	Starting in Junos OS Release 17.2R1, you can upgrade the solid-state drive (SSD) firmware on MX Series routers with the RE-S-X6-64G and RE-MX2K-X8-64G Routing Engines, on QFX10002-60C switches with the RE-QFX10002-60C Routing Engines, and PTX10002-60C routers with the RE-PTX10002-60C Routing Engines.

Upgrading the i40e NVM Firmware on Routing Engines with VM Host Support

Starting in Junos OS Release 19.3R1, in order to install VM Host image based on Linux WRL9, you have to upgrade the i40e NVM firmware to version 6.01. [Table 24 on page 423](#) lists the Junos OS releases which support i40e NVM firmware upgrade.

Starting in Junos OS Release 21.4 R1, in order to install VM Host image based on Linux WR LTS19, you have to upgrade the i40e NVM firmware to version 7.0 or later. . However, if the Junos OS version is upgraded to 21.4R1 or later using the standard Junos OS upgrade process, the i40e NVM firmware upgrade is done automatically during the upgrade process. The corresponding i40e driver version to support i40e NVM version 7.0 will be version 2.8.43 (or later).

Starting in Junos OS Release 22.3R1, when you upgrade your device from Junos OS version 18.2R1 to 22.3R1, the i40e NVM firmware is automatically upgraded from version 4.26 to version 7.0

Starting in Junos OS Release 24.1R1, in order to install VM Host image based on Linux WR LTS22, you have to upgrade the i40e NVM firmware to version 9.1 or later.



NOTE: i40e NVM version 6.01 is the prerequisite to install a LTS19 based image, else image installation will fail.



NOTE: In Junos OS Release 24.1R1 and later, ssh is disabled by default on all the routers with VM Host support (except SRX).

The i40e NVM firmware downgrade is not supported if the Junos OS version running on the system is downgraded from Junos OS 21.4R1. Hence, it is required to install a Junos OS version which supports the corresponding i40e NVM firmware version installed in the Routing Engine.

Table 1 lists the Junos OS releases contain both i40e driver version 2.8.43 and i40e driver version 2.4.3 to support both i40e NVM firmware version 6.01 and i40e NVM firmware version 7.0. Hence, they could be used on the Routing Engine which is running i40e NVM firmware version 6.01 or 7.0.

For JUNOS images prior to the listed versions below, i40e NVM firmware version 7.0 is not supported.

Table 23: Junos OS Releases that Support i40e NVM Firmware Version 6.01 and i40e NVM Firmware Version 7.0.

19.3	19.4	20.1	20.2	20.3	20.4	21.1+
19.3R2-S6	19.4R1-S4	20.1R2-S2	20.2R2-S3	20.3R1-S2	20.4R1-S1	21.1R1 and higher
19.3R3-S2	19.4R2-S4	20.1R3	20.2R3	20.3R2	20.4R2	
	19.4R3-S2			20.3R2-S1		
	19.4R3-S3			20.3R3		
				20.3X75-D10		

Table 24 on page 423 lists the Junos OS releases which support i40e NVM firmware upgrade.

Table 24: Junos OS Releases that Support i40e NVM Firmware Upgrade

Platform				18.x	19.x	21.x
EX9208				18.1R1 / 18.2R1 / 18.3R1 / 18.4R1	19.1R1 or later	
PTX5000				18.1R1 / 18.2R1 / 18.3R1 / 18.4R1	19.1R1 or later	
PTX3000				18.2R3 / 18.3R3 / 18.4R2	19.1R2 / 19.2R1	
MX240 / MX480 / MX960 / MX2010 / MX2020				18.1R1 / 18.2R1 / 18.3R1 / 18.4R1	19.1R1 or later	
MX2008				18.2R3 / 18.3R3 / 18.4R2	19.1R2 / 19.2R1	
MX10016/ MX10008 PTX10016/ MX10008				18.2R3 / 18.3R3 / 18.4R2	19.1R2 / 19.2R1	
SRX5400/ SRX5600/ SRX5800 (Supported only on SRX5K-RE3-128G)				Not applicable	Not applicable	21.4



NOTE: The following Junos OS Releases support automatic i40e NVM firmware upgrade from version 4.26 to version 7.00. This allows you to upgrade devices directly from older Junos OS Releases like 18.2 to Junos OS Releases 21.4 and higher.

- 21.4R3-S5 and higher
- 22.1R3-S3 and higher
- 22.2R3-S1 and higher
- 22.3R3 and higher
- 22.4R3 and higher
- 23.1R2 and higher
- 23.2R2 and higher
- 23.3R1 and higher
- 23.4R1 and higher

If you have to downgrade from any of the above mentioned releases, you can downgrade only to release mentioned in [Table 23 on page 422](#)

You can install older Junos OS images on the Routing Engine with an upgraded i40e NVM firmware as it supports i40e 2.4.3 driver versions. If you install an older version of the VM Host image, which is not listed in [Table 25 on page 424](#), using USB, the Routing Engine does not start up properly. In such a case, you can reinstall the VMHost image with a version which supports the new i40e NVM firmware.

[Table 25 on page 424](#) lists the platforms with NVM-6.01 firmware support.

Table 25: Junos OS Versions that Support i40e 2.4.3 Driver Versions

Platform	i40e-2.4.3/ NVM-6.01 Support					
EX9208					18.1R1 / 18.2R1 / 18.3R1 / 18.4R1	19.1R1 or later
PTX5000					18.1R1 / 18.2R3 / 18.3R1 / 18.4R1	19.1R1 or later

Table 25: Junos OS Versions that Support i40e 2.4.3 Driver Versions *(Continued)*

Platform		i40e-2.4.3/ NVM-6.01 Support				
PTX3000					18.2R3-S8 / 18.2R3 / 18.3R3 / 18.4R2	19.1R2/ 19.2R1
MX240/MX480/ MX960					18.1R1 / 18.2R1 / 18.3R1 / 18.4R1	19.1R1 or later
MX2010/MX2020					18.1R1 / 18.2R1 / 18.3R1 / 18.4R1	19.1R1 or later
MX2008					18.1R1 / 18.2R1 / 18.3R1 / 18.4R1	19.1R1 or later
MX10016/MX10008 PTX10016/MX10008					18.2R1 / 18.3R1 / 18.4R1	19.1R1 or later

i40e-NVM upgrade is optional for the following platforms:

- MX10003
- PTX10002-XX
- QFX1000, QFX10002
- QFX5000



NOTE:

- You must upgrade i40e NVM firmware only under the direction of a Juniper Networks support representative. Once you upgrade the NVM firmware, a downgrade action is not supported. For latest update, you can refer <https://kb.juniper.net/>.
- You must implement this procedure with a router console access. Also, you have to perform power cycling of the routing Engine multiple times during the firmware upgrade process.

Before you begin upgrading the firmware, check the current firmware version of the i40e NVM.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 0	RE BIOS	0	0.53.1		OK
Routing Engine 1	RE BIOS	0	0.43	0.53	OK
Routing Engine 1	RE FPGA	1	28.0.0	41.0	OK
Routing Engine 1	RE SSD1	3	0.0.0		OK
Routing Engine 1	RE SSD2	3	0.0.0		OK
Routing Engine 1	RE i40e-NVM	7	4.26		OK

If the value of Current version is less than 6.01, then you can use the following procedure for the i40e NVM firmware upgrade.

To upgrade i40e NVM firmware on routers with single Routing Engine:

1. Upgrade the device with the Junos OS image version which supports i40e NVM firmware upgrade. See [Table 24 on page 423](#).

Copy and install the jfirmware-vmhost package to the device.

If the file has been obtained from JTAC, use FTP or SCP to load the firmware file on the device. Save the file in the /var/tmp directory.

```
user@host> request vmhost software add /var/tmp/jfirmware-vmhost-x86-64-19.2R1.tgz
```

2. Upgrade the NVM firmware.

To initiate the upgrade, use the following command:

```
user@host> request system firmware upgrade re i40nvm
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 1	RE i40e-NVM	7	4.26	6.01	OK

Perform indicated firmware upgrade ? [yes,no] (no) yes

Firmware upgrade initiated, use "show system firmware" after reboot to verify the firmware version

Monitor the upgrade status by using the `show system firmware` command. If the upgrade is initiated the output displays `PROGRAMMING (0%)` as the status. However, note that the status `PROGRAMMING (0%)` does not increment during the process.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 0	RE BIOS	0		0.53.1	OK
Routing Engine 1	RE BIOS	0		0.43	OK
Routing Engine 1	RE FPGA	1		28.0.0	OK
Routing Engine 1	RE SSD1	3		0.0.0	OK
Routing Engine 1	RE SSD2	3		0.0.0	OK
Routing Engine 1	RE i40e-NVM	7	4.26	6.01	PROGRAMMING (0%)

3. Reboot the device by using the `request vmhost reboot` command.

```
user@host> request vmhost reboot
```

4. Verify the progress of i40e NVM upgrade on the console. You may have to perform power recycle of the Routing Engine multiple times. When you are prompted for a power cycle on your console, use external power cycle for power cycling the Routing Engine.

The following message is displayed on the console prompting you to perform a power cycle:

```
"Please Power Cycle your system now and run the NVM update utility again to complete the
update. Failure to do so will result in an incomplete NVM update.
Upgrade complete please power reboot
You may notify to power reboot again after reboot if required"
```

5. After a successful upgrade, verify the version of the firmware.



NOTE: The Current version is displayed as 6.1 instead of 6.01.

```
user@host> show system firmware
```

Part	Type	Tag	Current	Available	Status
------	------	-----	---------	-----------	--------

		version	version	
Routing Engine 0 RE BIOS	0		0.53.1	OK
Routing Engine 1 RE BIOS	0		0.43	OK
Routing Engine 1 RE FPGA	1		28.0.0	OK
Routing Engine 1 RE SSD1	3		0.0.0	OK
Routing Engine 1 RE SSD2	3		0.0.0	OK
Routing Engine 1 RE i40e-NVM	7	6.1	6.01	OK



NOTE: In case, you have run the `request vmhost snapshot` command with a Junos OS image which does not support i40e NVM firmware upgrade, (if the SSD recovery snapshot has a Junos OS version older than the Junos OS versions mentioned in [Table 25 on page 424](#)) we recommend you to take a snapshot using the `request vmhost snapshot` command again. Hence, in case of a recovery process, the SSD recovery snapshot will have a Junos OS image which supports NVM 6.01.

On routers with dual Routing Engines, you must use the `request chassis cb (offline | online) slot slot-number` to power cycle the Routing Engine. Thereby, you can avoid using an external power cycler and avoid abrupt power cycling of backup RE, which may cause file system errors.

To upgrade i40e NVM firmware on routers with dual Routing Engines:



NOTE: You must disable GRES before proceeding with the upgrade procedure. However, if you disable GRES in the beginning of the procedure, the device needs more number of switchovers for upgrading both the Routing Engines. Hence, to reduce the number of switchovers, it is recommended to upgrade the secondary Routing Engine first and then upgrade the primary Routing Engine .

1. Upgrade the device with the Junos OS image version which supports i40e NVM firmware upgrade. See [Table 24 on page 423](#).

Copy and install the `jfirmware-vmhost` package to the device.

If the file has been obtained from JTAC, use FTP or SCP to load the firmware file on the device. Save the file in the `/var/tmp` directory.

```
user@host> request vmhost software add /var/tmp/jfirmware-vmhost-x86-64-19.2R1.tgz
```

2. Upgrade the NVM firmware.

To initiate the upgrade, use the following command:

```
user@host> request system firmware upgrade re i40nvm
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 1	RE i40e-NVM	7	4.26	6.01	OK

Perform indicated firmware upgrade ? [yes,no] (no) yes

Firmware upgrade initiated, use "show system firmware" after reboot to verify the firmware version

Monitor the upgrade status by using the `show system firmware` command. If the upgrade is initiated the output displays `PROGRAMMING (0%)` as the status. However, note that the status `PROGRAMMING (0%)` does not increment during the process.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 0	RE BIOS	0		0.53.1	OK
Routing Engine 1	RE BIOS	0		0.43	OK
Routing Engine 1	RE FPGA	1		28.0.0	OK
Routing Engine 1	RE SSD1	3		0.0.0	OK
Routing Engine 1	RE SSD2	3		0.0.0	OK
Routing Engine 1	RE i40e-NVM	7	4.26	6.01	PROGRAMMING (0%)

3. Switch to the backup Routing Engine by using the `request chassis routing-engine master switch` command to switch primary role to other RE (i.e, RE1).



NOTE: This step is necessary, because in Step 6 you have to power cycle the Routing Engine which is undergoing the NVM upgrade (i.e, RE0) from RE1

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

Toggle mastership between routing engines ? [yes,no] (no) yes

Resolving mastership...

Complete. The other routing engine becomes the master.

4. Reboot the device by using the request `vmhost reboot` command from the Routing Engine which is undergoing the NVM upgrade (i.e, RE0).

```
user@host> request vmhost reboot
```

5. Monitor the console output. You may have to perform power recycle of the Routing Engine multiple times. When you are prompted for a power cycle on your console, use external power cycle for power cycling the Routing Engine. Or, you can use the command `request chassis cb slot slot offline` as described in Step 6.

The following message is displayed on the console prompting you to perform a power cycle:

```
"Please Power Cycle your system now and run the NVM update utility again to complete the
update. Failure to do so will result in an incomplete NVM update.
Upgrade complete please power reboot
You may notify to power reboot again after reboot if required"
```

6. From RE1, power cycle the RE0 using following command `request chassis cb slot slot offline`.

To power off RE0, use the command `request chassis cb slot 0 offline` and to power on RE0, use the command `request chassis cb slot 0 online`.

```
user@host> request chassis cb slot 0 offline
Offline initiated, use "show chassis environment cb" to verify
```

```
user@host> request chassis cb slot 0 online
Online initiated, use "show chassis environment cb" to verify
```

7. After a successful upgrade, verify the version of the firmware.



NOTE: The Current version is displayed as 6.1 instead of 6.01.

```
user@host> show system firmware
```

Part	Type	Tag	Current version	Available version	Status
Routing Engine 0	RE BIOS	0		0.53.1	OK
Routing Engine 1	RE BIOS	0		0.43	OK

Routing Engine 1 RE FPGA	1	28.0.0	OK
Routing Engine 1 RE SSD1	3	0.0.0	OK
Routing Engine 1 RE SSD2	3	0.0.0	OK
Routing Engine 1 RE i40e-NVM 7	6.1	6.01	OK



NOTE: In case, you have run the request vmhost snapshot command with a Junos OS image which does not support i40e NVM firmware upgrade, (if the SSD recovery snapshot has a Junos OS version older than the Junos OS versions mentioned in [Table 25 on page 424](#)) we recommend you to take a snapshot using the request vmhost snapshot command again. Hence, in case of a recovery process, the SSD recovery snapshot will have a Junos OS image which supports NVM 6.01.

8. Similarly, while upgrading i40e NVM on RE1, power cycle RE1 from RE0.

```
user@host> request chassis cb slot 1 offline
Offline initiated, use "show chassis environment cb" to verify
```

```
user@host> request chassis cb slot 1 online
Online initiated, use "show chassis environment cb" to verify
```

9. You need to perform this step only if the image you downloaded does not contain debugfs based mechanism to stop LLDP.

Download the lldp-patch-for-i40e-upgrade.tgz package. Copy and install the file in the /var/tmp/ directory on each Routing Engine.

```
user@host> request vmhost software add /var/tmp/lldp-patch-for-i40e-upgrade.tgz
Verified lldp-patch-for-i40e-upgrade signed by PackageDevelopmentEc_2018 method
ECDSA256+SHA256
[ re_name = RE-PTX-2X00x8 ]
Pushing script(s) to host ...
Install the script(s) under host-os....
Script(s) copy done
```

```
user@host> show version | match lldp
lldp-patch-for-i40e-upgrade
```

- 10. Reboot the device by using the `request vmhost reboot` command. The upgrade process is complete when the Routing Engine comes back online.

```
user@host> request vmhost reboot
```

Change History Table

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

Release	Description
19.3R1	Starting in Junos OS Release 19.3R1, in order to install VM Host image based on Linux WRL9, you have to upgrade the i40e NVM firmware to version 6.01.

Disable Autorecovery on Routing Engines with VM Host Support

IN THIS CHAPTER

- [Disabling Autorecovery on Routing Engines with VM Host Support | 433](#)

Disabling Autorecovery on Routing Engines with VM Host Support

The autorecovery feature helps recover the Junos OS automatically in the event of a corruption, thereby ensuring that the Junos OS is available for operations and management always. The host Junos OS tries to launch the Junos VM from the image available on the primary disk. However, if the guest Junos OS fails to launch, the host OS attempts to launch the Junos VM from the snapshot of the host OS image and Junos OS image available in the backup disk, provided request `vmhost snapshot` was the last operation performed. If the backup disk does not contain the snapshot, the host OS attempts to launch the Junos VM from the software available in the alternate set in the primary disk, provided request `vmhost upgrade` was the last operation performed.

The autorecovery feature is enabled by default on the guest Junos OS. For debugging purposes, if you do not want the host to recover the Junos VM automatically, you can disable the auto-recovery by the host.

To disable the guest auto-recovery, include the `no-auto-recovery` statement at the `[edit vmhost]` hierarchy level:

```
[edit vmhost]
no-auto-recovery
```

RELATED DOCUMENTATION

| *vmhost*

Personality Upgrade

IN THIS CHAPTER

- [Personality Upgrade Process | 434](#)
- [Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices | 446](#)

Personality Upgrade Process

IN THIS SECTION

- [Understanding the Personality Upgrade Process for a Device | 434](#)
- [Supported Personality Upgrades | 436](#)
- [Upgrading the Personality of a Device by Using a USB Flash Drive | 437](#)
- [Upgrading the Personality of a Device by Using CLI | 438](#)
- [Upgrading the Personality of a Device by Using a PXE Boot Server | 441](#)

Understanding the Personality Upgrade Process for a Device

IN THIS SECTION

- [Benefits of Personality Upgrade | 435](#)
- [Guidelines and Restrictions | 436](#)

Personality of a device can be defined as a combination of the purpose of the device and the solution that the device provides. For instance, a switch is a Layer 2 (Data Link Layer) device that is designed to

connect two or more networking devices on a network. Most switches (except a few Layer 3 switches) act as bridge devices that receive data packets from a source device, process the data, and forward it to a destination device within the same network. A router, in contrast, connects multiple networks. It is typically a Layer 3 (Network Layer) device because its primary function is to forward packets destined either for its own network or for other networks.

Starting in Junos OS Release 18.2R1, you can upgrade the personality of a device from the installed personality to a new personality without having to upgrade the entire device.

The build image loaded on a device defines the personality of the device. For instance, suppose you purchase a core router such as a PTX10008. The build image loaded on the router indicates its installed personality—that is, PTX10008. You can upgrade its personality and use it as an edge router such as an MX10008, in which case the device personality changes to MX10008. Similarly, you can change the device personality from an MX10008 to a PTX10008. You can also alternate between a switch (for example, QFX10002-60C) and a router (say, PTX10002-60C) by simply upgrading the device personality.

To upgrade the device personality from one device to another, you need certain common hardware components supported by both the devices. In the case of an MX10008 and a PTX10008, the presence of the common Routing and Control Board (RCB)—JNP10K-RE1—and the eight-slot universal chassis—JNP10008—enables you to upgrade from one device to the other seamlessly.



NOTE: When you order a spare JNP10K-RE1 RCB, the image of the MX10008 build is installed on that RCB. The spare JNP10K-RE1 also contains an image of the PTX10008 build at the `/var/tmp` location. You can upgrade an MX10008 router to an PTX10008 by using that image.

Benefits of Personality Upgrade

- **Reuse**—The same device (universal chassis) can be used as an edge router or a core router or a switch.
- **Time-saving**—You can quickly deploy the new device personality in the network.
- **Lower capital expenditure and operating costs**—You can upgrade the device personality instead of purchasing a new device.
- **Network Growth management**—Upgrading the personality of your device helps you manage the network growth when growth forecasts are discouraging.
- **Lower inventory and storage costs for distributors.**

Guidelines and Restrictions

This section describes the guidelines to consider when you upgrade the personality of a device:

- If you attempt to upgrade the personality of the device without using the recommended CLI command, the device can become inaccessible and unstable.
- There are no in-built restrictions or checks to validate the image that you plan to install on the device.
- Verify that the installed image supports the required command to upgrade to the new personality. If it does not, upgrade to a later version of the image before you upgrade to the new personality.
- When you upgrade the personality of the device, the configuration present in the device is migrated to the new personality. This is similar to a Junos OS upgrade. Therefore, any configuration that is not supported on the new personality must be deleted before you upgrade the personality. If any unsupported configuration is retained in the device after it reboots with the new image, the device returns to the factory-default configuration.
- Retain the minimum configuration required on the device, so the management interface is accessible.



NOTE: Juniper Networks does not support using the `request vmhost software rollback` command to revert to the previously installed personality.

Supported Personality Upgrades

Table 26 on page 436 displays the various combinations of device personality upgrades that are supported by Junos OS.

Table 26: Supported Personality Upgrades on Junos OS

Installed Personality	New Personality	Initial Junos OS Release	Common HW Component
MX10008	PTX10008	18.2	Routing and Control Board (JNP10K-RE1)
PTX10008	MX10008	18.2	Routing and Control Board (JNP10K-RE1)
QFX10002-60C	PTX10002-60C	18.2	
PTX10002-60C	QFX10002-60C	18.2	

Table 26: Supported Personality Upgrades on Junos OS (Continued)

Installed Personality	New Personality	Initial Junos OS Release	Common HW Component
MX10016	PTX10016	18.4	Routing and Control Board (JNP10K-RE1)
PTX10016	MX10016	18.4	Routing and Control Board (JNP10K-RE1)

You can upgrade the personality of the device to a new personality by:

- Using the USB flash drive
- Using the Junos OS CLI
- Using the PXE boot server

Upgrading the Personality of a Device by Using a USB Flash Drive

The build image loaded on the device defines the personality of the device. You can change the personality of the device by upgrading it.

In a USB upgrade, the content of the SSDs are erased and the image is installed from the USB flash drive to both the primary and secondary disks. Based on the image used, the device comes up as a PTX10008 or an MX10008. This is irrespective of the previously installed personality of the device.



NOTE: When you order a spare JNP10K-RE1 RCB, the image of the MX10008 build is installed on that RCB. The spare RCB also contains an image of the PTX10008 build at the `/var/tmp` location. You can upgrade an MX10008 router to an PTX10008 by using that image.

To upgrade the personality of the device by using a USB flash drive:

1. Insert the external USB flash drive. The external flash drive is detected.
2. Reboot the device.

```
user@host# run request vmhost reboot usb
OR
user@host# run request vmhost reboot
```

3. When prompted, unplug the USB flash drive after the system reboots.



NOTE: Juniper Networks does not support using the `request vmhost software rollback` command to revert to the previously installed personality.

Upgrading the Personality of a Device by Using CLI

IN THIS SECTION

- [How to Upgrade the Personality of a Device on Junos OS | 438](#)

The build image loaded on the device defines the personality of the device. You can change the personality of the device by upgrading it.

You can upgrade the personality of the device by using CLI configuration on devices running Junos OS.



NOTE: When you order a spare JNP10K-RE1 RCB, the image of the MX10008 build is installed on that RCB. The spare RCB also contains an image of the PTX10008 build at the `/var/tmp` location. You can upgrade an MX10008 router to an PTX10008 by using that image.

How to Upgrade the Personality of a Device on Junos OS

Use the following CLI procedure to upgrade the personality of a device running Junos OS.

- Verify that the installed image supports the required CLI command to upgrade to the new personality. If it does not, upgrade to a later version of the image before you upgrade to the new personality.
- Delete any configuration that is not supported or is not compatible with the new personality before you upgrade the personality. If any unsupported configuration is retained in the device after it reboots with the new image, the device returns to the factory-default configuration.

To upgrade the device to a new personality by using the Junos OS CLI:

1. In operational mode, verify the installed personality of the device. If you have purchased an MX10008 device, the installed personality of the device is displayed as `mx10008`. If you have purchased a PTX10008 device, the installed personality of the device is displayed as `ptx10008`.

```
user@host> show version
```

```
Hostname: host
Model: mx10008
```

```
Hostname: host
Model: ptx10008
```

2. Download the software package or build image from <https://www.juniper.net/support/>. For information about downloading software packages, see "Downloading Software (Junos OS)" on page 73. Save the software package to the `/var/path/package-name` directory on the router. For example, you can save the software package to the `/var/tmp` directory.



NOTE: Download the software package specific to the personality you want to upgrade to. The software package for PTX Series routers is different from the software package for MX Series routers.

3. In configuration mode, install the software package by using the `request vmhost software add path/package-name` command. Install the software package based on the new personality you want to upgrade to, as follows:

```
user@host# run request vmhost software add /var/tmp/junos-vmhost-install-ptx-x86-64-xyz.tgz
upgrade-to-model ptx10008 no-validate
```

```
user@host# run request vmhost software add /var/tmp/junos-vmhost-install-mx-x86-64-zyx.tgz
upgrade-to-model mx10008 no-validate
```



NOTE: If you do not specify the `no-validate` option, the router displays the following error message: **error: Upgrading to a different model is supported only with no-validate option.**

4. Reboot the router so the new package is loaded.

```
user@host# run request vmhost reboot
```

5. Run the `show version` command to verify that the upgrade is successful. If you have upgraded the personality of the device to an MX10008 device, the new personality of the device is displayed as `mx10008`. If you have upgraded the personality of the device to a PTX10008 device, the new personality of the device is displayed as `ptx10008`.

```
user@host> show version
```

```
Hostname: host
Model: ptx10008
```

```
Hostname: host
Model: mx10008
```



NOTE: Juniper Networks does not support using the `request vmhost software rollback` command to revert to the previously installed personality.

To ensure that all four partitions are upgraded to the same personality, follow these steps:

1. Boot from the solid-state drive (SSD) Disk 2 by using the `request vmhost reboot disk2` command.

```
user@host> request vmhost reboot disk2
```

2. Upgrade to the new personality by using the `upgrade-to-model` and `no-validate` options. This command upgrades both partitions on the SSD Disk 1.

```
user@host# run request vmhost software add junos-vmhost-install-x.tgz upgrade-to-model X no-validate reboot
```

If you are upgrading to PTX10008, include the package for the PTX Series routers and replace **X** with `ptx10008` before the `no-validate` option. If you are upgrading to MX10008, include the package for the MX Series routers and replace **X** with `mx10008` before the `no-validate` option.

3. After the device boots up from SSD Disk 1, take a snapshot from SSD Disk 1 to Disk 2.

```
user@host> request vmhost snapshot partition
```

This step ensures that both partitions on Disk 2 are upgraded to the new personality.

After you complete Step 1 through Step 3, all four partitions are upgraded to new personality.

Upgrading the Personality of a Device by Using a PXE Boot Server

The build image loaded on the device defines the personality of the device. You can change the personality of the device by upgrading it.

You can upgrade the personality of a device by using the Preboot Execution Environment (PXE) boot server. A PXE boot prepares a client/server environment to boot devices by using a network interface that is independent of available data storage devices or installed operating systems. The image of the operating system is stored on a TFTP server. You can have a separate PXE boot server for each image.



NOTE: When you order a spare JNP10K-RE1 RCB, the image of the MX10008 build is installed on that RCB. The spare RCB also contains an image of the PTX10008 build at the `/var/tmp` location. You can upgrade an MX10008 router to an PTX10008 by using that image.

To upgrade the personality of a device from the installed personality to the new personality by using the PXE boot server method:

- Copy the image you want installed on the device to the PXE boot server.
- Reboot the device to install the image.



NOTE: If you have already copied the image to the PXE boot server, reboot the device to install the image.

To copy the image you want installed to the PXE boot server and install the image:

1. Copy the downloaded installation media to the `/var/tmp` directory in the PXE boot server.

For example:

```
scp /volume/build/junos/18.2/release/zyx/ship/junos-vmhost-install-net-x86-64-xyz.tgz
user@host:/var/tmp/
```

2. Log in to the PXE boot server and verify the installation file.

For example:

```
user@host> ls -lh junos-vmhost-install-net-x86-64-xyz.tgz
-rw-r--r-- 1 root root 1.8G May 24 00:42 junos-vmhost-install-net-x86-64-xyz.tgz
```

3. Extract the **junos-vmhost-install-net** TAR file.

For example:

```
user@host> tar xvzf junos-vmhost-install-net-x86-64-xyz.tgz -C /var/tmp
attributes
junos-vmhost-install-ptx.tgz
manifest
manifest.certs
manifest.ecerts
manifest.esig
manifest.sig
package.xml
pkg_add_vmhost.sh
vmhost-install-net-x86_64-xyz.tgz
```

4. Remove the previously installed files, if any, from the **/tftpboot** directory.

```
user@host> rm -f /tftpboot
user@host> mkdir /tftpboot
```

5. Extract the network installation package.

For example:

```
user@host> tar xvzf /var/tmp/vmhost-install-net-x86_64-xyz.tgz -C /tftpboot/
./
./vmhost-version.sh
./bootpxe64.efi
./vmhost-version
./grub.cfg
..
...
-rw-rw-r-- 1 930 930 45M Oct 20 01:51 vmhost-install-net-x86_64-xyz.tgz
-rw-rw-r-- 1 930 930 6 Oct 20 01:51 vmhost-version
-rwxrwxr-x 1 930 930 416 Oct 20 01:51 vmhost-version.sh
```

```
-rw-r--r-- 1 930 930 6.9M Oct 20 01:51 vmlinuz
```

6. Rename or delete the previously installed root file **system/scripts** from the **/var/install** directory. Create a new **/var/install** directory.

```
user@host>mv /var/install /var/install_old
user@host>mkdir /var/install
```

7. Extract the installation package.
For example, this sample output is specific to the PTX Series device installation package.

```
user@host>tar xvzf /var/tmp/junos-vmhost-install-ptx.tgz -C /var/install
./
./vmhost-pkgs-version
./vm/
./vm/note
./vm/grub.cfg.ngre
./vm/vsmartd-1.0-0.x86_64.rpm
./vm/re_fpga-1.0-0.x86_64.rpm
./vm/veccd-1.0-0.x86_64.rpm
./vmhost-version.sh
./vmhost/
./vmhost/vmhost-x86_64-xyz.img.gz
...
...
./junos/junos-mtre-upgrade.sh
./vmhost-core-x86_64-15.1I20151019_1021_builder.tgz
./junos/
./junos/junos-install-x86-64-xyz.img.gz
```

8. Verify that the **/var/install** folder contains the **attributes** file. If the file does not exist in the specified location, copy the attribute file.



NOTE: The attribute file consists of the personality information of the image. If the attributes file is not present, the device is unable to upgrade to the new personality even when the PXE boot server has the relevant image.

```
user@host> mv /var/tmp/attributes /var/install
```

9. Set permissions for the files in the `/var/install` and `/tftpboot` directories.

```
user@host> chown root:root /tftpboot/*
user@host> chown -R root:root /var/install
user@host> chmod -R a+rw /var/install
```

10. Exit the PXE boot server.

```
user@host> exit
```

11. After you copy the image to the PXE boot server, to install the image on the device, reboot the device to install the image.

```
user@host> request vmhost reboot network
```

The router boots from the PXE server and installs the image on both the SSDs.

If the device fails to reboot, you can use the USB disk installation option. However, after using USB disk installation, if the router fails to reboot or is not accessible, follow these steps on the console:


1. Power cycle the chassis or remove the RCB (JNP10K-RE1) and plug it back in.
2. Press the **ESC** button to go to the Boot Manager Menu.
3. Select Boot Manager, and then press Enter.
4. Select the **ETH00 (xx:xx:xx:xx:xx:xx)** option. A warning message is displayed. At the prompt, select **y** to install the image on both the primary and secondary disks.

```
WARNING: The installation will erase the contents of your disks.
Install vmhost and Junos Software on Primary and Secondary disk [y/n] y
```

5. In operational mode, verify that the upgrade is successful. If you have upgraded the personality of the device to an MX10008, the new personality of the device is `mx10008`. If you have upgraded the personality of the device to a PTX10008, the new personality of the device is `ptx10008`.

```
user@host> show version
Hostname: host
Model: ptx10008
```

```
user@host> show version
Hostname: host
Model: mx10008
```



NOTE: Juniper Networks does not support using the `request vmhost software rollback` command to revert to the previously installed personality.

Change History Table

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

Release	Description
18.2R1	Starting in Junos OS Release 18.2R1, you can upgrade the personality of a device from the installed personality to a new personality without having to upgrade the entire device.

RELATED DOCUMENTATION

-
- [Upgrading the Personality of QFX10002-60C and PTX10002-60C Switches Using the PXE Boot Server](#)

Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices

IN THIS SECTION

- [Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the PXE Boot Server | 446](#)
- [Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the USB Option | 450](#)
- [Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the CLI Option | 451](#)
- [Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using Zero Touch Provisioning \(ZTP\) | 454](#)

The installed image on your devices determines the personality of the device. Juniper Networks offers benefits of changing the personality of your device. You can install the image of QFX10002-60C in PTX10002-60C device and vice versa. You can install the new personality via Preboot Execution Environment (PXE) boot method boot, USB, CLI, and ZTP.

Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the PXE Boot Server

You can configure your QFX10002-60C device as a PTX10002-60C device or your PTX10002-60C device as a QFX10002-60C device. The image loaded on the device determines the personality of the device. For example, if you have purchased a QFX10002-60C device, the installed personality is a QFX10002-60C device. When you upgrade the QFX10002-60C device to a PTX10002-60C device, the new personality of the device is a PTX10002-60C device. Similarly, if you have purchased a PTX10002-60C device, the installed personality is a PTX10002-60C device and the new personality, if you upgrade, is a QFX10002-60C device.

You can install the new personality on the devices using Preboot Execution Environment (PXE) boot method. A PXE boot prepares a client/server environment to boot devices by using a network interface that is independent of available data storage devices or installed operating systems. The image of the operating system is stored on a TFTP server. You can have separate PXE boot servers for each image.



NOTE: When you upgrade the QFX10002-60C personality to a PTX10002-60C personality, the QFX10002-60C default configuration is deleted, and the PTX10002-60C configuration becomes the default configuration. When you upgrade the PTX10002-60C personality to a QFX10002-60C personality, the PTX10002-60C default configuration is deleted, and the QFX10002-60C configuration becomes the default

configuration. Additionally, the software snapshot in the secondary disk is deleted, and the new software snapshot is installed in the secondary disk. For example, if you upgrade the QFX10002-60C personality to a PTX10002-60C personality, the QFX10002-60C snapshot is deleted, and the PTX10002-60C snapshot is installed in the secondary disk.



NOTE: When you order the spare JNP10002-60C-CHAS, it is preloaded with the QFX10002-60C and PTX10002-60C software images in the **/var/tmp** location. If you want to convert a QFX10002-60C device to a PTX10002-60C device, use the PTX10002-60C image. If you want to convert a PTX10002-60C device to QFX10002-60C device, use the QFX10002-60C image.

For example, to upgrade the QFX10002-60C device from the installed personality of QFX10002-60C to the new personality of PTX10002-60C device using the PXE Boot Server Option:

- Copy the image you want installed on the QFX10002-60C device to the PXE Boot Server.
- Reboot the device to install the image.



NOTE: If you have already copied the image to the PXE Boot server, reboot the device to install the image.

To copy the image you want installed to the PXE Boot Server:

1. Copy the downloaded installation media to the **/var/tmp** directory in the PXE boot server.

For example:

```
scp /volume/build/junos/18.2/release/zyx/ship/junos-vmhost-install-ptx-x86-64-xyz.tgz
user@host:/var/tmp/
```

2. Log in to the PXE boot server and verify the installation file.

```
user@host> ls -lh junos-vmhost-install-ptx-x86-64-xyz.tgz
-rw-r--r-- 1 root root 1.8G May 24 00:42 junos-vmhost-install-net-x86-64-xyz.tgz
```

3. Extract the **junos-vmhost-install-net** TAR file.

```
user@host> tar xvzf junos-vmhost-install-ptx-x86-64-xyz.tgz -C /var/tmp
attributes
junos-vmhost-install-ptx.tgz
```

```
manifest
manifest.certs
manifest.ecerts
manifest.esig
manifest.sig
package.xml
pkg_add_vmhost.sh
vmhost-install-net-x86_64-xyz.tgz
```

4. Remove the previously installed files, if any, from the **/tftpboot** directory.

```
user@host> rm -f /tftpboot
user@host> mkdir /tftpboot
```

5. Extract the network installation package.

```
user@host> tar xvzf /var/tmp/junos-vmhost-install-ptx-x86-64-xyz.tgz -C /tftpboot/
./
./vmhost-version.sh
./bootpxe64.efi
./vmhost-version
./grub.cfg
..
...
-rw-rw-r-- 1 930 930 45M Oct 20 01:51 vmhost-install-net-x86_64-xyz.tgz
-rw-rw-r-- 1 930 930 6 Oct 20 01:51 vmhost-version
-rwxrwxr-x 1 930 930 416 Oct 20 01:51 vmhost-version.sh
-rw-r--r-- 1 930 930 6.9M Oct 20 01:51 vmlinuz
```

6. Rename or delete the previously installed root file system/scripts from the **/var/install** directory. Create a new **/var/install** directory.

```
user@host> mv /var/install /var/install_old
user@host> mkdir /var/install
```

7. Extract the installation package.

```
user@host> tar xvzf /var/tmp/junos-vmhost-install-ptx-x86-64.tgz -C /var/install
./
./vmhost-pkgs-version
```



```

./vm/
./vm/note
./vm/grub.cfg.ngre
./vm/vsmartd-1.0-0.x86_64.rpm
./vm/re_fpga-1.0-0.x86_64.rpm
./vm/veccd-1.0-0.x86_64.rpm
./vmhost-version.sh
./vmhost/
./vmhost/vmhost-x86_64-xyz.img.gz
...
...
./junos/junos-mtre-upgrade.sh
./vmhost-core-x86_64-15.1I20151019_1021_builder.tgz
./junos/
./junos/junos-install-x86-64-xyz.img.gz

```

8. Verify that the **/var/install** folder contains the **attributes** file. If the file does not exist in the specified location, copy the attribute file.



NOTE: The attribute file consists of the personality information of the image. If the attributes file is not present, the device is unable to upgrade to the new personality even when the PXE boot server has the relevant image.

```

user@host> mv /var/tmp/attributes /var/install

```

9. Set permissions for the files in the **/var/install** and **/tftpboot** directories.

```

user@host> chown root:root /tftpboot/*
user@host> chmod a+rwX /tftpboot/*
user@host> chown -R root:root /var/install
user@host> chmod -R a+rwX /var/install

```

10. Exit the PXE boot server.

```

user@host> exit

```

After you copy the image to the PXE Boot Server, to install the image on the device, reboot the device to install the image. You can use the request `vmhost reboot network` command to install the image. The device boots from the PXE server and installs the image on both the SSDs. However, if the device fails

to reboot, you can use the USB disk installation option. If the device fails to reboot or is not accessible, follow these steps:

1. Power cycle the device.
2. Press the **ESC** button to go to the Boot Manager Menu.
3. Select Boot Manager, and then press Enter.
4. Select **ETH00 (xx:xx:xx:xx:xx:xx)** option. A warning message is displayed. At the prompt, select **y** to install the image on both the primary and secondary disks.

```
WARNING: The installation will erase the contents of your disks.
Install vmhost and Junos Software on Primary and Secondary disk [y/n]
y
```

5. In operational mode, verify that the upgrade is successful.

```
user@host> show version
Hostname: host
Model: ptx10002-60C
```



NOTE: Juniper Networks does not support using the request vmhost software rollback command to revert to the previously installed personality.

Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the USB Option

You can configure your QFX10002-60C device as a PTX10002-60C device or your PTX10002-60C device as a QFX10002-60C device. The image loaded on the device determines the personality of the device. For example, if you have purchased a QFX10002-60C device, the installed personality is a QFX10002-60C device. When you upgrade the QFX10002-60C device to a PTX10002-60C device, the new personality of the device is a PTX10002-60 C device. Similarly, if you have purchased a PTX10002-60C device, the installed personality is a PTX10002-60C device and the new personality, if you upgrade, is a QFX10002-60C device.

In an USB upgrade, the content of the SSDs are erased and the image is installed from the USB to both the primary and secondary disks. Based on the image used, the device comes up as either a QFX10002-60C or a PTX10002-60C device. This is irrespective of the previously installed personality of the JNP10002-60C-CHAS chassis.



NOTE: When you order the spare JNP10002-60C-CHAS, it is preloaded with the QFX10002-60C and PTX10002-60C software images in the `/var/tmp` location. If you want to convert a QFX10002-60C device to a PTX10002-60C device, use the PTX10002-60C image. If you want to convert a PTX10002-60C device to QFX10002-60C device, use the QFX10002-60C image.

For example, to upgrade the QFX10002-60C device from the installed personality of QFX10002-60C to the new personality of PTX10002-60C device using the USB Option:

1. Insert the external USB flash drive. The external flash drive is detected.
2. Reboot the device.

```
user@host# run request vmhost reboot usb
OR
user@host# run request vmhost reboot
```

3. Unplug the USB flash drive after the system reboots, when prompted.



NOTE: Juniper Networks does not support using the `request vmhost software rollback` command to revert to the previously installed personality.

Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using the CLI Option

You can configure your QFX10002-60C device as a PTX10002-60C device or your PTX10002-60C device as a QFX10002-60C device. The image loaded on the device determines the personality of the device. For example, if you have purchased a QFX10002-60C device, the installed personality is a QFX10002-60C device. When you upgrade the QFX10002-60C device to a PTX10002-60C device, the new personality of the device is a PTX10002-60C device. Similarly, if you have purchased a PTX10002-60C device, the installed personality is PTX10002-60C and the new personality, if you upgrade, is a QFX10002-60C device.



NOTE: When you upgrade the QFX10002-60C personality to a PTX10002-60C personality, the QFX10002-60C default configuration is deleted, and the PTX10002-60C configuration becomes the default configuration. When you upgrade the PTX10002-60C personality to a QFX10002-60C personality, the PTX10002-60C default configuration is deleted, and the QFX10002-60C configuration becomes the default configuration. Additionally, the software snapshot in the secondary disk is deleted, and

the new software snapshot is installed in the secondary disk. For example, if you upgrade the QFX10002-60C personality to a PTX10002-60C personality, the QFX10002-60C snapshot is deleted, and the PTX10002-60C snapshot is installed in the secondary disk.



NOTE: When you order the spare JNP10002-60C-CHAS, it is preloaded with the QFX10002-60C and PTX10002-60C software images in the `/var/tmp` location. If you want to convert a QFX10002-60C device to a PTX10002-60C device, use the PTX10002-60C image. If you want to convert a PTX10002-60C device to QFX10002-60C device, use the QFX10002-60C image.

- Verify if the installed image supports the required command to upgrade to the new personality. If it does not, upgrade to a later version of the image before you upgrade to the new personality.
- Delete any configuration that is not supported or not compatible with the new personality before you upgrade the personality. If any unsupported configuration is retained in the device after it reboots with the new image, the device returns to the factory-default configuration.

For example, to upgrade the QFX10002-60C device from the installed personality of QFX10002-60C to the new personality of PTX10002-60C device using the CLI Option:

1. In operational mode, verify the installed personality of the device

```
user@host> show version
```

```
Hostname: host
Model: QFX10002-60C
```

2. Download the software package from <https://www.juniper.net/support/>. For information about downloading software packages, see "Downloading Software (Junos OS)" on page 73. Save the software package to the `/var/path/package-name` directory on the device. For example, you can save the software package to the `/var/tmp` directory.



NOTE: Download the software package specific to the personality you want to upgrade to. The software package for QFX Series devices is different from the software package for the PTX Series devices.

3. In operational mode, install the software package by using the request `vmhost software add path/package-name` command. For example, to install the `junos-vmhost-install-ptx-x86-64-zyx.tgz` package:

```
user@host> request vmhost software add /var/tmp/junos-vmhost-install-ptx-x86-64-zyx.tgz no-validate
```



NOTE: If you do not specify the `no-validate` option, the device displays the following error message **error: Upgrading to a different model is supported only with no-validate option .**

4. Run the `show version` command to verify that the upgrade is successful.

```
user@host> show version
```

```
Hostname: host
Model: ptx10002-60C
```



NOTE: Juniper Networks does not support using the `request vmhost software rollback` command to revert to the previously installed personality.

To ensure that all 4 partitions are upgraded to the same personality, follow these steps:

1. Boot from solid-state drive (SSD) Disk 2 using the `request vmhost reboot disk2` command.

```
user@host> request vmhost reboot disk2
```

2. Upgrade to the new personality using `no-validate` option. This command upgrades both partitions on SSD Disk 1.

```
user@host> request vmhost software add package-name no-validate reboot
```

For example:

```
user@host> request vmhost software add junos-vmhost-install-ptx-x86-64-zyx.tgz no-validate reboot
```

If you are upgrading to the PTX10002-60C device, include the package for the PTX10002-60C device. If you are upgrading to the QFX10002-60C device, include the package for the QFX10002-60C device.

3. After booting up from SSD1, take a snapshot from SSD1 to SSD2.

```
user@host> request vmhost snapshot partition
```

This ensures that both partitions on SSD2 are upgraded to new personality.

Upgrading the Personality of QFX10002-60C and PTX10002-60C Devices Using Zero Touch Provisioning (ZTP)

You can configure your QFX10002-60C device as a PTX10002-60C device or your PTX10002-60C device as a QFX10002-60C device. The image loaded on the device determines the personality of the device. For example, if you have purchased a QFX10002-60C device, the installed personality is a QFX10002-60C device. When you upgrade the QFX10002-60C device to a PTX10002-60C device, the new personality of the device is a PTX10002-60C device. Similarly, if you have purchased a PTX10002-60C device, the installed personality is PTX10002-60C and the new personality, if you upgrade, is a QFX10002-60C device.



NOTE: When you upgrade the QFX10002-60C personality to a PTX10002-60C personality, the QFX10002-60C default configuration is deleted, and the PTX10002-60C configuration becomes the default configuration. When you upgrade the PTX10002-60C personality to a QFX10002-60C personality, the PTX10002-60C default configuration is deleted, and the QFX10002-60C configuration becomes the default configuration. If you have provided your own Junos OS configuration, that configuration becomes the default configuration. Additionally, the software snapshot in the secondary disk is deleted, and the new software snapshot is installed in the secondary disk. For example, if you upgrade the QFX10002-60C personality to a PTX10002-60C personality, the QFX10002-60C snapshot is deleted, and the PTX10002-60C snapshot is installed in the secondary disk.



NOTE: When you order the spare JNP10002-60C-CHAS, it is preloaded with the QFX10002-60C and PTX10002-60C software images in the `/var/tmp` location. If you want to convert a QFX10002-60C device to a PTX10002-60C device, use the PTX10002-60C image. If you want to convert a PTX10002-60C device to QFX10002-60C device, use the QFX10002-60C image.

- Verify if the installed image supports the required command to upgrade to the new personality. If it does not, upgrade to a later version of the image before you upgrade to the new personality.

- Delete any configuration that is not supported or not compatible with the new personality before you upgrade the personality. If any unsupported configuration is retained in the device after it reboots with the new image, the device returns to the factory-default configuration.

Before you begin:

- Ensure that the switch or router has access to the following network resources:
 - The DHCP server that provides the location of the software image and configuration files on the network

Refer to your DHCP server documentation for configuration instructions.

- The File Transfer Protocol (anonymous FTP), Hypertext Transfer Protocol (HTTP), or Trivial File Transfer Protocol (TFTP) server on which the software image and configuration files are stored



NOTE: Although TFTP is supported, we recommend that you use FTP or HTTP instead, because these transport protocols are more reliable.



CAUTION: HTTP URLs are limited to 256 characters in length.

- A Domain Name System (DNS) server to perform reverse DNS lookup
- (Optional) An NTP server to perform time synchronization on the network
- (Optional) A system log (syslog) server to manage system log messages and alerts
- Locate and record the MAC address printed on the switch or router chassis.



CAUTION: You cannot commit a configuration while the switch or router is performing the software update process. If you commit a configuration while the switch or router is performing the configuration file autoinstallation process, the process stops, and the configuration file is not downloaded from the network.

For example, to upgrade the QFX10002-60C device from the installed personality of QFX10002-60C to the new personality of PTX10002-60C device using ZTP:

1. In operational mode, verify the installed personality of the device

```
user@host> show version
```

```
Hostname: host
Model: QFX10002-60C
```

2. Boot the device.
3. Make sure the device has the default factory configuration installed.
Issue the request `vmhost zeroize` command on the device that you want to provision.
4. Download the software package specific to the personality you want to upgrade from <https://www.juniper.net/support/>.
The software package for QFX Series devices is different from the software package for the PTX Series devices.
5. Save the software package and the configuration file to the FTP, HTTP, or TFTP server from which the device will download these files.
6. Configure the DHCP server to provide the necessary information to the switch or router.
Configure IP address assignment.

You can configure dynamic or static IP address assignment for the management address of the switch or router. To determine the management MAC address for static IP address mapping, add 1 to the last byte of the MAC address of the switch or router, which you noted before you began this procedure.

7. Define the format of the vendor-specific information for DHCP option 43 in the **dhcpd.conf** file.
Here is an example of an ISC DHCP 4.2 server **dhcpd.conf** file:

```
option space NEW_OP; option;
option NEW_OP.image-file-name code 0 = text;
option NEW_OP.config-file-name code 1 = text;
option NEW_OP.image-file-type code 2 = text;
option NEW_OP.transfer-mode code 3 = text;
option NEW_OP.alt-image-file-name code 4 = text;
option NEW_OP.jloader-file code 5 = text;
option NEW_OP-encapsulation code 43 = encapsulate NEW_OP;
```




NOTE: Starting in Junos OS Release 18.2R1, a new DHCP option is introduced to set the timeout value for the file downloads over FTP. If the `transfer-mode` is set as FTP, the default value for the timeout is automatically set as 120 minutes, that is, in case the FTP session gets interrupted due to loss of connectivity in the middle of a file transfer, it will timeout after 120 minutes and ZTP will attempt to retry the file fetching process. This value can be overridden using the DHCP option as follows:

```
option NEW_OP.ftp-timeout code 7 = text;
option NEW_OP.ftp-timeout "val";
```

where "val" is the user configurable timeout value in seconds and must be provided within quotes (like, "val").

8. Configure the following DHCP option 43 suboptions:



NOTE: DHCP option 43 suboptions 05 through 255 are reserved.

- Suboption 00: The name of the software image file to install.



NOTE: When the DHCP server cannot use suboption 00, configure the software image filename using suboption 04. If both suboption 00 and suboption 4 are defined, suboption 04 is ignored.

```
option NEW_OP.image-file-name "/dist/images/jinstall-ex-4200-13.2R1.1-domestic-signed.tgz";
```

- Suboption 01: The name of the script or configuration file to install.

```
option NEW_OP.config-file-name "/dist/config/jn-switch35.config";
```

The following list provides the types of scripts and their associated interpreter paths:

- Shell script interpreter path: `#!/bin/sh`
- SLAX script interpreter path: `#!/usr/libexec/ui/cscript`
- Python script interpreter path: `#!/usr/bin/python`

Unsigned Python scripts are only supported on limited platforms, such as the QFX5100 device. If you try to execute unsigned Python scripts on devices that do not provide support, error messages will be issued.



NOTE: If the file does not contain special characters (!#) , ZTP determines that the file is a configuration file and loads the configuration file.

- Suboption 02: The symbolic link to the software image file to install.

```
option NEW_OP.image-file-type "symlink";
```



NOTE: If you do not specify suboption 2, the ZTP process handles the software image as a filename, not a symbolic link.

- Suboption 03: The transfer mode that the switch or router uses to access the TFTP, FTP, or HTTP server. If you select FTP as the transfer mode, Junos OS uses the anonymous FTP login to download files from the FTP server.

```
option NEW_OP.transfer-mode "ftp";
```



NOTE: If suboption 03 is not configured, TFTP becomes the transfer mode by default.

- Suboption 04: The name of the software image file to install.



NOTE: When the DHCP server cannot use suboption 00, configure the image file using suboption 04. If both suboption 00 and suboption 4 are defined, suboption 04 is ignored.

```
option NEW_OP.alt-image-file-name "/dist/images/jinstall-ex-4200-13.2R1.1-domestic-signed.tgz";
```

- Suboption 05: The name of the JLoader image file to install.

```
NEW_OP.jloader-file "jloader-qfx-5-14.1X53-D26-signed.tgz";
```

9. (Mandatory) Configure either option 150 or option 66.



NOTE: You must configure either option 150 or option 66. If you configure both option 150 and option 66, option 150 takes precedence, and option 66 is ignored. Also, make sure you specify an IP address, not a hostname, because name resolution is not supported.

- Configure DHCP option 150 to specify the IP address of the FTP, HTTP, or TFTP server.

```
option option-150 code 150={ip-address};
option option-150 10.100.31.71;
```

- Configure DHCP option 66 to specify the IP address of the FTP, HTTP, or TFTP server.

```
option tftp-server-name "10.100.31.71";
```

10. (Optional) Configure DHCP option 7 to specify one or more system log (syslog) servers.

```
option log-servers 10.100.31.72;
```

11. (Optional) Configure DHCP option 42 to specify one or more NTP servers.

```
option ntp-servers 10.100.31.73;
```

12. (Optional) Configure DHCP option 12 to specify the hostname of the switch or router.

```
option hostname "jn-switch35";
```

The following sample configuration shows the DHCP options you just configured:

```
host jn-switch35 {
    hardware ethernet ac:4b:c8:29:5d:02;
    fixed-address 10.100.31.36;
```

```

option tftp-server-name "10.100.31.71";

option host-name "jn-switch35";
option log-servers 10.100.31.72;
option ntp-servers 10.100.31.73;
option NEW_OP.image-file-name "/dist/images/jinstall-ex-4200-13.2R1.1-domestic-
signed.tgz";
option NEW_OP.transfer-mode "ftp";
option NEW_OP.config-file-name "/dist/config/jn-switch35.config";
option NEW_OP.jloader-file "jloader-qfx-5-14.1X53-D26-signed.tgz";
}

```

Based on the DHCP options you just configured, the following statements are appended to the Junos OS configuration file (for example, jn-switch35.config):

```

system {
    host-name jn-switch35;

    syslog {
        host 10.100.31.72 {
            any any;
        }
    }
    ntp {
        server 10.100.31.73;
    }
}

```

13. Monitor the ZTP process by looking at the following log files.



NOTE: When SLAX (live operating system based on Linux) scripts are issued, the `op-script.log` and `event-script.log` files are produced.

- `/var/log/dhcp_logfile`
- `/var/log/event-script.log`
- `/var/log/image_load_log`
- `/var/log/messages`

- /var/log/op-script.log
- /var/log/script_output

You can also monitor the ZTP process by looking at error messages and issuing operational commands. See "[Monitoring Zero Touch Provisioning](#)" on page 506 for more information.

14. Run the `show version` command to verify that the upgrade is successful.

```
user@host> show version
```

```
Hostname: host  
Model: ptx10002-60C
```

7

PART

Zero Touch Provisioning and Secure Zero Touch Provisioning

- [Configure Zero Touch Provisioning and Secure Zero Touch Provisioning | 463](#)
-

Configure Zero Touch Provisioning and Secure Zero Touch Provisioning

IN THIS CHAPTER

- [Zero Touch Provisioning | 463](#)
- [Secure Zero Touch Provisioning | 519](#)
- [Switching between Secure Zero Touch Provisioning and Zero Touch Provisioning | 530](#)

Zero Touch Provisioning

IN THIS SECTION

- [Zero Touch Provisioning Overview | 463](#)
- [Zero Touch Provisioning Using DHCP Options | 468](#)
- [Zero Touch Provisioning Using DHCPv6 Options | 489](#)
- [Zero Touch Provisioning on SRX Series Firewalls | 496](#)
- [Monitoring Zero Touch Provisioning | 506](#)

Zero Touch Provisioning installs or upgrades the software automatically on your new Juniper Networks devices with minimal manual intervention.

Zero Touch Provisioning Overview

IN THIS SECTION

- [ZTP Workflow | 464](#)

- [Provisioning a Device Using a Script | 465](#)
- [Zero Touch Provisioning Restart Process Triggers | 466](#)
- [Caveats Relating to ZTP | 467](#)
- [Zero Touch Provisioning Using WAN Interfaces on PTX1000 Routers | 468](#)

Zero Touch Provisioning (ZTP) allows you to provision new Juniper Networks devices in your network automatically, with minimal manual intervention. You can use either management ports or network ports, depending on your device, to connect to the network. When you physically connect a device to the network and boot it with a default factory configuration, the device upgrades (or downgrades) the software release and autoinstalls a configuration file from the network. The configuration file can be a configuration or a script. Using scripts, you can create device-specific configuration files and perform HTTP request operations to web servers to download specific configuration files or software releases.

To locate the necessary software image and configuration files on the network, the device uses information that you have configured on a Dynamic Host Configuration Protocol (DHCP) server. If you do not configure the DHCP server to provide this information, the device boots with the preinstalled software and default factory configuration.

For certain switches, you can use the phone-home client (PHC) to provision software for the switch. When the switch boots up, if there are DHCP options that have been received from the DHCP server for ZTP, ZTP resumes. If DHCP options are not present, PHC is attempted. For more information about PHC, see "[Provision a Virtual Chassis Using the Phone-Home Client](#)" on page 541.



NOTE: To see which platforms support ZTP, in a browser, go to [Feature Explorer](#). In the **Explore Features** section of the Feature Explorer page, select **All Features**. In the **Features Grouped by Feature Family** box, select Zero Touch Provisioning. You can also type the name of the feature in the **Search for Features** edit box. See the Release History Table at the end of this topic for more details of how ZTP support has expanded.

ZTP Workflow

When a device boots up with the default configuration, the following events take place:

1. DHCP client is run on supported interfaces.
2. DHCP server provisions an IP address and includes several DHCP options in the reply related to the ZTP process.

3. The device processes the DHCP options and locates configuration files, executes scripts, and upgrades and/or downgrades software.
4. If both the image and configuration files are present, the image is installed and the configuration is applied.
5. If only the image file is present, the image is installed on the device.
6. If the image is the same as the image already installed on the device, ZTP continues and skips the installation step.
7. If the image was unable to be fetched by the device, ZTP will try to fetch the image again.
8. If the image is corrupted, installation fails.

If installation fails for any reason, ZTP will restart.

9. If only the configuration file is present, the configuration is downloaded.

If the first line of the file consists of the `#!` characters followed by an interpreter path, then the file is considered a script, and the script is executed by the interpreter. If the script returns an error, ZTP state machine will re-fetch the script and attempt to execute the script again.

If the configuration file is unable to be downloaded, the ZTP process will try to download it again.

If the configuration file is corrupted, has syntax errors, or includes commands that are unsupported by the device, the device will be unable to commit, and the retry mechanism will restart.

10. If there is no image or configuration file, the ZTP process starts again.
11. If there is no file server information, the ZTP process starts again.
12. Once the configuration is committed, the ZTP process is deemed successful and terminates.

Provisioning a Device Using a Script

During the ZTP process, when you connect and boot a new networking device, the device requests an IP address from the DHCP server. The server provides the IP address, and if configured, the filenames and locations for the software image and configuration file for the device. The configuration file can be a configuration or a script.

If a configuration file is provided, the operating system determines if the file is a script based on the first line of the file. If the first line contains the characters `#!` followed by an interpreter path, the operating system treats the file as a script and executes it with the specified interpreter.

If the script returns an error (that is, a nonzero value), the ZTP state machine re-fetches the script and attempts to execute it again. This continues until the script executes successfully.

Table 27 on page 466 outlines the supported script types, the corresponding interpreter path, and the platforms that support that script type during the ZTP process.

Table 27: Scripts Supported During ZTP

Script Type	Interpreter Path	Platform Support
Shell script	<code>#!/bin/sh</code>	All devices
SLAX script	<code>#!/usr/libexec/ui/cscript</code>	All devices
Python script	<code>#!/usr/bin/python</code>	Devices running Junos OS with Enhanced Automation Devices running Junos OS Evolved



NOTE: For security reasons, Junos OS has strict requirements for running unsigned Python scripts on devices running Junos OS. Only devices running Junos OS with Enhanced Automation and devices running Junos OS Evolved support using unsigned Python scripts in DHCP option 43 suboption 01.

If the operating system does not find the characters `#!` followed by an interpreter path, it treats the file as a configuration in text format and loads the configuration on the device.

Zero Touch Provisioning Restart Process Triggers

ZTP restarts when any of the following events occur:

- Request for configuration file, script file, or image file fails.
- Configuration file is incorrect, and commit fails.
- No configuration file and no image file is available.
- Image file is corrupted, and installation fails.
- No file server information is available.
- DHCP server does not have valid ZTP parameters configured.
- When none of the DHCP client interfaces goes to a bound state.
- ZTP transaction fails after six attempts to fetch configuration file or image file.

When any of these events occur, ZTP resets the DHCP client state machine on all of the DHCP client-configured interfaces (management and network) and then restarts the state machine. Restarting the state machine enables the DHCP client to get the latest DHCP server-configured parameters.

Before ZTP restarts, approximately 15 to 30 seconds must elapse to allow enough time to build a list of bound and unbound DHCP client interfaces.

The list of bound and unbound DHCP client interfaces can contain:

- No entries.
- Multiple DHCP client interfaces.

Priority is given to the DHCP client interfaces that have received all ZTP parameters (software image file, configuration file, and file server information) from the DHCP server.

After the lists of bound and unbound client interfaces are created, and a DHCP client gets selected for ZTP activity, any existing default route is deleted and the DHCP client interface that was selected adds a new default route. In order to add a new default route, only one ZTP instance can be active.

After ZTP restarts, the DHCP client attempts fetching files from the DHCP server for up to six times, with ten to fifteen seconds elapsing between attempts. Every attempt, whether successful or not, is logged and can be seen on the console.

If there is a failure, or the number of attempts exceeds the limit, ZTP stops. ZTP then clears the DHCP client bindings and restarts the state machine on the DHCP-configured interfaces.

The ZTP restart process continues until there is either a successful software upgrade, or an operator manually commits a user configuration and deletes the ZTP configuration.

Caveats Relating to ZTP

There are two downgrade limitations for EX Series switches:

- If you downgrade to a software version earlier than Junos OS Release 12.2, in which ZTP is not supported, the configuration file autoinstall phase of the zero touch provisioning process does not happen.
- To downgrade to a software version that does not support resilient dual-root partitions (Junos OS Release 10.4R2 or earlier), you must perform some manual work on the device. For more information, see ["Configuring Dual-Root Partitions" on page 161](#).

The following are caveats for QFX Series switches:

- On QFX3500 and QFX3600 switches running the original CLI, you cannot use ZTP to upgrade from Junos OS Release 12.2 or later to Junos OS Release 13.2X51-D15 or later.

- QFX5200 switches only work with HTTP in 15.1X53-D30. FTP and TFTP protocols are not supported.
- If you are performing Zero Touch Provisioning (ZTP) with a Junos OS image that contains enhanced automation for the QFX5100 switch, configure root authentication, and the provider name, license type, and deployment scope for Chef and Puppet at the `[edit system]` hierarchy in the configuration file that is fetched from the server:

```
{ master:0}
root# set root-authentication (encrypted-password password | plain-text-password password |
ssh-dsa public-key | ssh-rsa public-key)
root# set extensions providers juniper license-type customer deployment-scope commercial
root# set extensions providers chef license-type customer deployment-scope commercial
```

- In Junos OS Release 18.1R1, if you are upgrading the software, you must perform a full software upgrade. A full upgrade includes upgrading both the Junos OS software and the host software packages.

Zero Touch Provisioning Using WAN Interfaces on PTX1000 Routers

Zero Touch Provisioning (ZTP) allows you to provision your router in your network automatically, with minimal manual intervention. Starting in Junos OS Release 19.3R1, you can use either WAN interfaces or management interfaces, to automatically download and install the appropriate software and the configuration file on your router during the ZTP bootstrap process.

When you connect the router to the network at the first time, you can choose any available WAN port on the router to connect the optics. The ZTP automatically configures WAN interfaces based on the optics type, and then connects your device to the Dynamic Host Configuration Protocol (DHCP) server to perform the bootstrap process.

The WAN interfaces created based on the optics type you connected to the device and the WAN interface speed auto-transitions through all possible supported port speeds until the ZTP gets completed successfully. The speed auto-transition ensures to establish physical link of the WAN port with the optics you connected and the peer end device connectivity to the DHCP server.

The [PTX1000 Packet Transport Router Hardware Guide](#) shows the available combinations for the ports on the PTX1000 routers.

Zero Touch Provisioning Using DHCP Options

Zero Touch Provisioning (ZTP) allows for automatic provisioning of Juniper Network devices that you add to your network. You can provision any supported device by using either a script to be executed or a configuration file to be loaded. You will also need to configure a DHCP server with required information, which is provided in this procedure, to use ZTP.

Optionally, you can configure an HTTP proxy server for either the phone-home server or redirect server. When the phone-home client receives information regarding the HTTP proxy server via DHCP option 43 suboption 8, it will create an HTTPS transparent tunnel with the proxy server. Once the tunnel is established, the phone-home client uses the tunnel as a proxy for the phone-home server or redirect server. The phone-home client downloads the software image and configuration file through the tunnel onto the device. Once bootstrapping is complete, the device reboots and the tunnel quits.

ZTP requires that your device is in a factory default state. The device from the factory boots with preinstalled software and factory default configuration. On a device that does not currently have the factory default configuration, you can issue the `request system zeroize` command.



NOTE: The `request system zeroize` command is not supported on PTX1000, PTX10001-20C, QFX10002-60C, PTX10002-60C devices. You must issue the `request vmhost zeroize` command (instead of `request system zeroize`) for factory default configuration on PTX1000 routers.



NOTE: On PTX10001-20C devices, after you issue the `request vmhost zeroize` command, you will see the following message twice: VMHost Zeroization : Erase all data, including configuration and log files ? [yes,no] (no) yes warning: Vmhost will reboot and may not boot without configuration Erase all data, including configuration and log files? [yes,no] (no) yes

Before you begin:

- Ensure that the device has access to the following network resources:
 - The DHCP server that provides the location of the software image and configuration files on the network

Refer to your DHCP server documentation for configuration instructions.

- The File Transfer Protocol (anonymous FTP), Hypertext Transfer Protocol (HTTP), or Hypertext Transfer Protocol Secure (HTTPS), or Trivial File Transfer Protocol (TFTP) server on which the software image and configuration files are stored



NOTE: Although TFTP is supported, we recommend that you use FTP or HTTP instead, because these transport protocols are more reliable.



CAUTION: HTTP URLs are limited to 256 characters in length.

- A Domain Name System (DNS) server to perform reverse DNS lookup (not supported).
- (Optional) An NTP server to perform time synchronization on the network
- (Optional) A system log (syslog) server to manage system log messages and alerts.

Syslog messages will be forwarded to this syslog server during ZTP.

- (Optional) An HTTP proxy server for either the phone-home server or redirect server.
- Locate and record the MAC address for your device.

On PTX10008 devices, the management MAC addresses are located on routing engines.



CAUTION: You cannot commit a configuration while the device is performing the software update process. If you commit a configuration while the device is performing the configuration file autoinstallation process, the process stops, and the configuration file is not downloaded from the network.

To enable zero touch provisioning for a device using DHCP options:

1. Boot the device.
2. Make sure the device has the default factory configuration installed.

Issue the request `system zeroize` command on the device that you want to provision.



NOTE: The request `system zeroize` command is not supported on PTX1000 devices. You must issue the request `vmhost zeroize` command (instead of request `system zeroize`) for factory default configuration on PTX1000 devices.

We recommend you provision the DHCP server and save the software and configuration file in the specified DHCP server path on the file server.

3. Download the software image file and/or the configuration file to the FTP, HTTP, or TFTP server from which the device will download these files.



NOTE: If you are performing zero touch provisioning with a Junos OS image that contains enhanced automation for the QFX5100 device, configure root authentication and the provider name, license type, and deployment scope for Chef and Puppet at the `[edit system]` hierarchy in the configuration file that is fetched from the server:

```
{ master:0}
root# set root-authentication (encrypted-password password | plain-text-password
password | ssh-dsa public-key | ssh-rsa public-key)
root# set extensions providers juniper license-type customer deployment-scope
commercial
root# set extensions providers chef license-type customer deployment-scope
commercial
```

4. Configure the DHCP server to provide the necessary information to the device.

Configure IP address assignment.

You can configure the dynamic or static IP address assignment for the management address of the device.

To determine the management MAC address for static IP address mapping, add 1 to the last byte of the MAC address of the device, which you noted before you began this procedure.



NOTE: This address can be any address from the pool.

5. Define the format of the vendor-specific information for DHCP option 43 in the **dhcpcd.conf** file.

Here is an example of an ISC DHCP 4.2 server **dhcpcd.conf** file:

```
option space NEW_OP; option;
option NEW_OP.image-file-name code 0 = text;
option NEW_OP.config-file-name code 1 = text;
option NEW_OP.image-file-type code 2 = text;
option NEW_OP.transfer-mode code 3 = text;
option NEW_OP.alt-image-file-name code 4= text;
option NEW_OP.http-port code 5= text;
option NEW_OP-encapsulation code 43 = encapsulate NEW_OP;

option NEW_OP.proxyv4-info code 8 = text;
```

Here is an example of a Kea DHCP 2.4.1 server **kea-dhcp4.conf** file:

```
{
  "Dhcp4": {
    "option-def": [
      {
        "name": "image-file-name",
```

```

        "code": 0,
        "space": "vendor-encapsulated-options-space",
        "type": "string",
        "record-types": "",
        "array": false,
        "encapsulate": ""
    },
    {
        "name": "config-file-name",
        "code": 1,
        "space": "vendor-encapsulated-options-space",
        "type": "string",
        "record-types": "",
        "array": false,
        "encapsulate": ""
    },
    {
        "name": "image-file-type",
        "code": 2,
        "space": "vendor-encapsulated-options-space",
        "type": "string",
        "record-types": "",
        "array": false,
        "encapsulate": ""
    },
    {
        "name": "transfer-mode",
        "code": 3,
        "space": "vendor-encapsulated-options-space",
        "type": "string",
        "record-types": "",
        "array": false,
        "encapsulate": ""
    },
    {
        "name": "alt-image-file-name",
        "code": 4,
        "space": "vendor-encapsulated-options-space",
        "type": "string",
        "record-types": "",
        "array": false,
        "encapsulate": ""
    }
},

```



```

    {
      "name": "http-port",
      "code": 5,
      "space": "vendor-encapsulated-options-space",
      "type": "string",
      "record-types": "",
      "array": false,
      "encapsulate": ""
    },
    {
      "name": "proxyv4-info",
      "code": 8,
      "space": "vendor-encapsulated-options-space",
      "type": "string",
      "record-types": "",
      "array": false,
      "encapsulate": ""
    }
  ]
}

```



NOTE: Starting in Junos OS Release 18.2R1, a new DHCP option is introduced to set the timeout value for the file downloads over FTP. If the `transfer-mode` is set as FTP, the default value for the timeout is automatically set as 120 minutes, that is, in case the FTP session gets interrupted due to loss of connectivity in the middle of a file transfer, it will timeout after 120 minutes and ZTP will attempt to retry the file fetching process. This value can be overridden using the following DHCP option.:

Here is an example of an ISC DHCP 4.2 server `dhcpcd.conf` file:

```

option NEW_OP.ftp-timeout code 7 = text;
option NEW_OP.ftp-timeout "val";

```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

Definition:

```
{
  "Dhcp4": {
    "option-def": [
      {
        "name": "ftp-timeout",
        "code": 7,
        "space": "vendor-encapsulated-options-space",
        "type": "string",
        "record-types": "",
        "array": false,
        "encapsulate": ""
      }
    ]
  }
}
```

Assignment:

```
"option-data": [
  {
    "name": "ftp-timeout",
    "space": "vendor-encapsulated-options-space",
    "data": "val"
  }
]
```

where “val” is the user configurable timeout value in seconds and must be provided within quotes (like, “val”).

6. Configure the following DHCP option 43 suboptions:

- Suboption 00: The name of the software image file to install.



NOTE: When the DHCP server cannot use suboption 00, configure the software image filename using suboption 04. If both suboption 00 and suboption 04 are defined, suboption 04 is ignored.

Here is an example of an ISC DHCP 4.2 server dhcpd.conf file:

```
option NEW_OP.image-file-name "/dist/images/jinstall-ex-4200-13.2R1.1-domestic-signed.tgz";
```

Here is an example of a Kea DHCP 2.4.1 server kea-dhcp4.conf file:

```
"option-data": [
  {
    "name": "image-file-name",
    "space": "vendor-encapsulated-options-space",
    "data": "images/junos-install-ex-x86-64-23.4R2-S3.9.tgz"
  }
]
```

- Suboption 01: The name of the script or configuration file to install.

Here is an example of an ISC DHCP 4.2 server dhcpd.conf file:

```
option NEW_OP.config-file-name "/dist/config/jn-switch35.config";
```

Here is an example of a Kea DHCP 2.4.1 server kea-dhcp4.conf file:

```
"option-data": [
  {
    "name": "config-file-name",
    "space": "vendor-encapsulated-options-space",
    "data": "configs/ex4400-ZG4723350182-config.txt"
  }
]
```



NOTE: ZTP determines if the file is a script file based on the first line of the file. If the first line contains the characters `#!` followed by an interpreter path, ZTP treats the file as a script and executes it with the specified interpreter path. For a script to execute, the script file must provide the ability to fetch and load a valid configuration file on the device during the ZTP process.

The following list provides the types of scripts and their associated interpreter paths:

- Shell script interpreter path: `#!/bin/sh`
- SLAX script interpreter path: `#!/usr/libexec/ui/cscript`
- Python script interpreter path: `#!/usr/bin/python`

For security reasons, Junos OS has strict requirements for running unsigned Python scripts on devices running Junos OS. Only devices running Junos OS with Enhanced Automation and devices running Junos OS Evolved support running unsigned Python scripts as part of the ZTP process.

If the file does not contain special characters (`#!`), ZTP determines that the file is a configuration file and loads the configuration file.



NOTE: Starting in Junos OS Release 21.1R1, ZTP Python scripts that are fetched from the ZTP server should be migrated to use Python 3 because Python 2.7 is no longer supported. In other words, the interpreter directive line should point to Python 3 and also the script's code needs to be migrated to Python 3.

- Suboption 02: The symbolic link to the software image file to install.

Here is an example of an ISC DHCP 4.2 server `dhcpd.conf` file:

```
option NEW_OP.image-file-type "symlink";
```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

```
"option-data": [
  {
    "name": "image-file-type",
    "space": "vendor-encapsulated-options-space",
    "data": "symlink"
  }
]
```



NOTE: If you do not specify suboption 2, the ZTP process handles the image filename as a filename, not a symbolic link.

- Suboption 03: The transfer mode that the device uses to access the TFTP, FTP, HTTP, or HTTPS server. If you select FTP as the transfer mode, Junos OS uses the anonymous FTP login to download files from the FTP server.

Here is an example of an ISC DHCP 4.2 server `dhcpd.conf` file:

```
option NEW_OP.transfer-mode "ftp";
```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

```
"option-data": [
  {
    "name": "transfer-mode",
    "space": "vendor-encapsulated-options-space",
    "data": "ftp"
  }
]
```

- Suboption 04: The name of the software image file to install.



NOTE: If the DHCP server does not support suboption 00, configure the image file using suboption 04. If both suboption 00 and suboption 4 are defined, suboption 04 is ignored.

Here is an example of an ISC DHCP 4.2 server `dhcpd.conf` file:

```
option NEW_OP.alt-image-file-name "/dist/images/jinstall-ex-4200-13.2R1.1-domestic-
signed.tgz";
```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

```
"option-data": [
  {
    "name": "alt-image-file-name",
    "space": "vendor-encapsulated-options-space",
    "data": "images/junos-install-ex-x86-64-23.4R2-S3.9.tgz"
  }
]
```

- Suboption 05: The HTTP port that the device uses to download either the image or configuration file or both instead of the default HTTP port.

Here is an example of an ISC DHCP 4.2 server `dhcpd.conf` file:

```
option NEW_OP.http-port code 5= 8080;
```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

```
"option-data": [
  {
    "name": "http-port",
    "space": "vendor-encapsulated-options-space",
    "data": "8080"
  }
]
```

- Suboption 08: HTTP proxy server information that is passed from the DHCP server to the DHCP client. This is useful when the device needs to access the phone-home server or redirect server via a proxy server.



NOTE: When you configure the DHCP server and HTTP proxy server, make sure that you use the correct port number to allow traffic to flow through the secure tunnel. Also, make sure that the hostname or IP address of the HTTP proxy server and port number are separated by a colon: for example, 192.168.10.10:8080. If you don't use a colon, port 1080 is used.

When the DHCP client receives the HTTP proxy server information, it is saved in the `/var/etc/phc_vendor_specific_info.xml` (INET) file.

If the DHCP client does not receive the HTTP proxy server information, nothing is saved to the `/var/etc/phc_vendor_specific_info.xml` (INET) file, and the DHCP client moves into a bound state.

You can renew the HTTP proxy server information by issuing the request `dhcp client renew interface` command. The DHCP client fetches the valid HTTP proxy server information from the DHCP server. Using the command is simpler than having to restart the provisioning process. When the HTTP proxy server is renewed, or the HTTP proxy server information is changed or deleted, `jdhcp` will rewrite the `/var/etc/phc_vendor_specific_info.xml` file with the latest information received from suboption 8.

Here is an example of an ISC DHCP 4.2 server dhcpd.conf file:

```
option NEW_OP.proxyv4-info code 8 = text;
```

Here is an example of a Kea DHCP 2.4.1 server kea-dhcp4.conf file:

```
{
  "Dhcp4": {
    "option-def": [
      {
        "name": "proxyv4-info",
        "code": 8,
        "space": "vendor-encapsulated-options-space",
        "type": "string",
        "record-types": "",
        "array": false,
        "encapsulate": ""
      }
    ]
  }
}
```

Here's the format for this option:

ISC DHCP 4.2 server dhcpd.conf file:

```
option NEW_OP.proxyv4-info "http://<proxyname>:<port-number>";
```

Kea DHCP 2.4.1 server kea-dhcp4.conf file:

```
"option-data": [
  {
    "name": "proxyv4-info",
    "space": "vendor-encapsulated-options-space",
    "data": "http://<proxyname>:<port-number>"
  }
]
```

Here's an example of the format using a fictitious proxy name:

ISC DHCP 4.2 server dhcpd.conf file:

```
option NEW_OP.proxyv4-info "http://test-mr2:3128";
```

Kea DHCP 2.4.1 server kea-dhcp4.conf file:

```
"option-data": [
  {
    "name": "proxyv4-info",
    "space": "vendor-encapsulated-options-space",
    "data": "http://test-mr2:3128"
  }
]
```

7. (Mandatory) Configure either option 150 or option 66.



NOTE: You must configure either option 150 or option 66. If you configure both option 150 and option 66, option 150 takes precedence, and option 66 is ignored. Also, make sure you specify an IP address, not a hostname, because name resolution is not supported.

- Configure DHCP option 150 to specify the IP address of the FTP, HTTP, HTTPS, or TFTP server.

Here is an example of an ISC DHCP 4.2 server dhcpd.conf file:

```
option option-150 code 150={ ip-address};
option option-150 10.100.31.71;
```

Here is an example of a Kea DHCP 2.4.1 server kea-dhcp4.conf file:

Definition:

```
{
  "Dhcp4": {
    "option-def": [
      {
        "name": "tftp-server-address",
        "code": 150,
        "space": "dhcp4",
        "type": "ipv4-address",
```



```

        "array": true
    }
]
}
}

```

Assignment:

```

"option-data": [
  {
    "name": "tftp-server-address",
    "data": "10.100.31.71"
  }
]

```

- Configure DHCP option 66 to specify the IP address of the FTP, HTTP, HTTPS, or TFTP server.

Here is an example of an ISC DHCP 4.2 server `dhcpd.conf` file:

```
option tftp-server-name "10.100.31.71";
```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

Definition:

```

{
  "Dhcp4": {
    "option-def": [
      {
        "name": "tftp-server-name",
        "code": 66,
        "space": "dhcp4",
        "type": "string",
        "array": false
      }
    ]
  }
}

```

Assignment:

```
"option-data": [
  {
    "name": "tftp-server-name",
    "data": "10.100.31.71"
  }
]
```

8. (Optional) Configure DHCP option 7 to specify one or more system log (syslog) servers.

Here is an example of an ISC DHCP 4.2 server `dhcpd.conf` file:

```
option log-servers 10.100.31.72;
```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

Definition:

```
{
  "Dhcp4": {
    "option-def": [
      {
        "name": "log-servers",
        "code": 7,
        "space": "dhcp4",
        "type": "ipv4-address",
        "array": true
      }
    ]
  }
}
```

Assignment:

```
"option-data": [
  {
    "name": "log-servers",
    "data": "10.100.31.72"
```

```
    }
  ]
}
```

9. (Optional) Configure DHCP option 42 to specify one or more NTP servers.

List each NTP server separated by a space.

Here is an example of an ISC DHCP 4.2 server `dhcpd.conf` file:

```
option ntp-servers 10.100.31.73;
```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

Definition:

```
{
  "Dhcp4": {
    "option-def": [
      {
        "name": "ntp-servers",
        "code": 42,
        "space": "dhcp4",
        "type": "ipv4-address",
        "array": true
      }
    ]
  }
}
```

Assignment:

```
"option-data": [
  {
    "name": "ntp-servers",
    "data": "10.100.31.73"
  }
]
```

10. (Optional) Configure DHCP option 12 to specify the hostname of the device.

For example:

```
option hostname "jn-switch35";
```

The following configuration shows an example of the DHCP options you just configured in this procedure:

Here is an example of an ISC DHCP 4.2 server `dhcpd.conf` file::

```
host jn-switch35 {
    hardware ethernet ac:4b:c8:29:5d:02;
    fixed-address 10.100.31.36;
    option tftp-server-name "10.100.31.71";
    option NEW_OP.ftp-timeout "val";
    option host-name "jn-switch35";
    option log-servers 10.100.31.72;
    option ntp-servers 10.100.31.73;
    option NEW_OP.image-file-name "/dist/images/jinstall-ex-4200-13.2R1.1-domestic-
signed.tgz";
    option NEW_OP.transfer-mode "ftp";
    option NEW_OP.http-port code 5= 8080;
    option NEW_OP.config-file-name "/dist/config/jn-switch35.config";
}
```

Here is an example of a Kea DHCP 2.4.1 server `kea-dhcp4.conf` file:

```
{
  "Dhcp4": {
    "client-classes": [
      {
        "name": "jn-switch35",
        "test": "pkt4.mac == 0xac4b:c829:5d02",
        "option-data": [
          {
            "name": "vendor-encapsulated-options"
          },
          {
            "name": "image-file-name",
            "space": "vendor-encapsulated-options-space",
            "data": "images/junos-install-ex-x86-64-23.4R2-S3.9.tgz"
          },
        ],
      },
    ],
  },
}
```

```

        {
            "name": "config-file-name",
            "space": "vendor-encapsulated-options-space",
            "data": "configs/ex4400-ZG4723350182-config.txt"
        },
        {
            "name": "image-file-type",
            "space": "vendor-encapsulated-options-space",
            "data": "symlink"
        },
        {
            "name": "transfer-mode",
            "space": "vendor-encapsulated-options-space",
            "data": "http"
        },
        {
            "name": "http-port",
            "space": "vendor-encapsulated-options-space",
            "data": "8080"
        },
        {
            "name": "ftp-timeout",
            "space": "vendor-encapsulated-options-space",
            "data": "90"
        },
        {
            "name": "tftp-server-address",
            "data": "10.1.60.103"
        },
        {
            "name": "log-servers",
            "data": "10.100.31.72"
        },
        {
            "name": "ntp-servers",
            "data": "10.100.31.73"
        }
    ]
}

```

Based on the DHCP options configured in this example, the following items are added to the [edit system] hierarchy:

```
system {
  host-name jn-switch35;

  syslog {
    host 10.100.31.72 {
      any any;
    }
  }
  ntp {
    server 10.100.31.73;
  }
}
```

Juniper devices running Junos OS send additional DHCP options to the DHCP server in the Discover packet during Zero Touch Provisioning (ZTP). The following options enable the DHCP server to identify specific devices based on parameters such as vendor, model name, and serial number, which may vary across various Juniper platforms.

DHCP option number		DHCP option name
Option 12		Hostname
Option 60		Vendor Class Identifier (VCI)

Model	DHCP Option 12 (Hostname)	DHCP Option 60 (VCI)
EX4400-24MP	<serial-number>	Juniper-ex4400-24mp- <serial-number>
MX10-T	<serial-number>	Juniper-mx10-t- <serial-number>
SRX320	Option not sent	Juniper-srx320

Here's an example of how to classify a device based on its specific serial number and assign a unique set of DHCP options for ZTP processing.

ISC DHCP 4.2 server dhcpd.conf file:

```
class "Juniper-ex4400-ZG4723350182" {
    match if substring (option vendor-class-identifier, 14, 12) = "ZG4723350182";
    option NEW_OP.image-file-name "images/junos-install-ex-x86-64-23.4R2-S3.9.tgz";
    option NEW_OP.config-file-name "configs/ex4400-ZG4723350182-config.txt";
    option NEW_OP.image-file-type "symlink";
    option NEW_OP.transfer-mode "http";
    option NEW_OP.http-port "8080";
    option OPTION-150 10.1.60.103;
}
```

Kea DHCP 2.4.1 server kea-dhcp4.conf file:

```
{
  "Dhcp4": {
    "client-classes": [
      {
        "name": "Juniper-ex4400-ZG4723350182",
        "test": "substring(option[60].hex,14,12) == 'ZG4723350182'",
        "option-data": [
          {
            "name": "vendor-encapsulated-options"
          },
          {
            "name": "image-file-name",
            "space": "vendor-encapsulated-options-space",
            "data": "images/junos-install-ex-x86-64-23.4R2-S3.9.tgz"
          },
          {
            "name": "config-file-name",
            "space": "vendor-encapsulated-options-space",
            "data": "configs/ex4400-ZG4723350182-config.txt"
          },
          {
            "name": "image-file-type",
            "space": "vendor-encapsulated-options-space",
            "data": "symlink"
          },
          {
            "name": "transfer-mode",
```


Zero Touch Provisioning Using DHCPv6 Options



NOTE: Zero Touch Provisioning (ZTP) using DHCPv6 options isn't supported on Junos OS Flex images. A Flex image has the word "flex" in the filename. Here is an example filename of a Flex image: `jinstall-host-qfx-5e-flex-x86-64-20.4R3.8-secure-signed.tgz`.

The DHCPv6 protocol doesn't have a subnet option for the IA_NA (identity association for non-temporary addresses) to learn and install subnet routes. Instead, the subnet route is installed through Neighbor Discovery Protocol.

In IPv6, devices periodically advertise IPv6 prefixes along with other link parameters using Router Advertisement (RA) messages. On the client (Juniper device running ZTP), once the DHCPv6 client is bound, the Neighbor Discovery Protocol (NDP) will learn these prefixes and installs the prefix routes via the client interface, with the next hop as the link to the local address of the gateway device.

On the client device, router advertisement configuration is enabled by default along with the DHCPv6 configuration.

- Ensure that the device has access to the following network resources:
 - The DHCP server that provides the location of the software image and configuration files on the network

Refer to your DHCP server documentation for configuration instructions.

- On the MX Series, the File Transfer Protocol (anonymous FTP), Trivial File Transfer Protocol (TFTP), Hypertext Transfer Protocol (HTTP), or Hypertext Transfer Protocol Secure (HTTPS) server on which the software image and configuration files are stored.



CAUTION: HTTP URLs are limited to 256 characters in length.

- On the EX3400, EX4300, QFX5100, and QFX5200 devices, the Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) server on which the software image and configuration files are stored.



CAUTION: HTTP URLs are limited to 256 characters in length.

- (Optional) An HTTP proxy server for either the phone-home server or redirect server.
- Locate and record the MAC address printed on the device.

Zero Touch Provisioning (ZTP) allows for automatic provisioning of Juniper Network devices that you add to your network. You can provision any supported device by using either a script to be executed or a configuration file to be loaded.

To use ZTP, you configure a DHCP server to provide the required information. If you do not configure the DHCP server to provide this information, the device boots with the preinstalled software and default factory configuration. If your device is not in a factory default state, you can issue the `request system zeroize` command.

Optionally, you can configure an HTTP proxy server for either the phone-home server or redirect server. When the phone-home client receives information regarding the HTTP proxy server via DHCP option 17 suboption 8, it will create an HTTPS transparent tunnel with the proxy server. Once the tunnel is established, the phone-home client uses the tunnel as a proxy for the phone-home server or redirect server. The phone-home client downloads the software image and configuration file through the tunnel onto the device. Once bootstrapping is complete, the device reboots and the tunnel quits.



NOTE: Starting in Junos OS Release 20.2R1-S1, the DHCPv6 client is supported the MX-Series, EX3400, EX4300, QFX5100, and QFX5200 switches. Both DHCPv4 and DHCPv6 clients are included as part of the default configuration. During the bootstrap process, the device first uses the DHCPv4 client to request for information regarding image and configuration file from the DHCP server. The device checks the DHCPv4 bindings sequentially. If there is a failure with one of the DHCPv4 bindings, the device will continue to check for bindings until provisioning is successful. If there are no DHCPv4 bindings, however, the device will check for DHCPv6 bindings and follow the same process as for DHCPv4 until the device can be provisioned successfully. The DHCP server uses DHCPv6 options 59 and 17 and applicable sub-options to exchange ZTP-related information between itself and the DHCP client.



CAUTION: You cannot commit a configuration while the device is performing the software update process. If you commit a configuration while the device is performing the configuration file autoinstallation process, the process stops, and the configuration file is not downloaded from the network.

To use zero touch provisioning for a device using DHCPv6 options:

1. Boot the device.
2. Make sure the device has the default factory configuration installed.
 - If multiple DHCP replies arrive, the ZTP chooses the best set of arguments.
 - If multiple interfaces provide the same arguments, ZTP chooses one of the equal interfaces.

- If there is an error while connecting to the DHCP server, ZTP tries again to connect to the DHCP server. If multiple interfaces again provide the same arguments, ZTP chooses one of the interfaces.

We recommend you to provision the DHCP server and save the software and configuration file in the specified DHCP server path on the file server.

3. Download the software image file and the configuration file to the FTP, HTTP, HTTPS, or TFTP server from which the device will download these files.
4. Configure the DHCP server to provide the necessary information to the device.
5. Configure IP address assignment.

You can configure dynamic or static IP address assignment for the management address of the device. To determine the management MAC address for static IP address mapping, add 1 to the last byte of the MAC address of the device, which you noted before you began this procedure.

6. Define the format of the DHCPv6 option 59 (OPT_BOOTFILE_URL) in the **dhcpcd6.conf** file, so the server can send information about URLs to images to the client.



NOTE: Only the HTTP and HTTPS transport protocols are supported on the EX3400, EX4300, QFX5100, and QFX5200 devices.

Here's the format for this option:

```
transfer-mode://[<ipv6-address>]:<port-number>/<path/image-file-name>
```

For example:

```
ftp://[2001:db8::40]:21/ZTP/bootimage.tgz
tftp://[2001:db8::40]:69/ZTP/bootimage.tgz
http://[2001:db8::40]:80/ZTP/bootimage.tgz
https://[2001:db8::40]:443/ZTP/bootimage.tgz
```

The transfer mode and IPv6 address are required, but the port number is optional. If you do not specify the port number, the default port number of the transfer mode is used. If you specify the port number in options 17 and 59, then the port number mentioned in option 17 vendor-specific information option is used.

You can specify the image file name in either option 59 or option 17. If the image file name is mentioned in both options 59 and 17, then the image name mentioned in option 17 vendor-specific information option is used.

7. Define the format of the vendor-specific information for the following DHCP option 17 suboptions:

Here is an example of an ISC DHCP 4.2 server `dhcdd6.conf` file:

```
option space NEW_OP_V6 code width 2 length width 2;
option NEW_OP_V6. image-file-type code 0 = text;
option NEW_OP_V6.config-file-name code 1 = text;
option NEW_OP_V6. image-file-type code 2 = text;
option NEW_OP_V6.transfer-mode code 3 = text;
option NEW_OP_V6. alt-image-file-name code 4 = text;
option NEW_OP_V6.port-number code 5 = text;
option NEW_OP_V6. jloader-file code 6 = text;
option NEW_OP_V6. ftp-timeout code 7 = text;
option NEW_OP_V6.proxyv6-info code 8 = text;
option vsio.NEW_OP_V6 code 2636 = encapsulate NEW_OP_V6;
```

- Suboption 00: The name of the software image file to install.



NOTE: When the DHCP server cannot use suboption 00, configure the software image filename using suboption 04. If both suboption 00 and suboption 4 are defined, suboption 04 is ignored.

For example:

```
option NEW_OP_V6.image-file-name "ZTP_IMAGES/jinstall-qfx-5-20.2-img.tgz";
```

- Suboption 01: The name of the script or configuration file to install.

For example:

```
option NEW_OP_V6.config-file-name "ZTP_FILES/baseline_config";
```



NOTE: ZTP determines if the file is a script file based on the first line of the file. If the first line contains the characters `#!` followed by an interpreter path, ZTP treats the file as a script and executes it with the specified interpreter path. In order for a script to execute, the script file must provide the ability to fetch and load a valid configuration file on the device during the ZTP process.

The following list provides the types of scripts and their associated interpreter paths:

- Shell script interpreter path: `#!/bin/sh`
- SLAX script interpreter path: `#!/usr/libexec/ui/cscript`
- Python script interpreter path: `#!/usr/bin/python`

For security reasons, Junos OS has strict requirements for running unsigned Python scripts on devices running Junos OS. Only devices running Junos OS with Enhanced Automation and devices running Junos OS Evolved support running unsigned Python scripts as part of the ZTP process.

If the file does not contain special characters (`#!`), ZTP determines that the file is a configuration file and loads the configuration file.



NOTE: Starting in Junos OS Release 21.1R1, ZTP Python scripts that are fetched from the ZTP server should be migrated to use Python 3 because Python 2.7 is no longer supported. In other words, the interpreter directive line should point to Python 3 and also the script's code needs to be migrated to Python 3.

- Suboption 02: The image type.

```
option NEW_OP_V6.image-file-type symlink;
```



NOTE: If you do not specify suboption 2, the ZTP process handles the software image as a filename, not a symbolic link.

- Suboption 04: The name of the software image file to install.



NOTE: When the DHCP server cannot use suboption 00, configure the image file using suboption 04. If both suboption 00 and suboption 4 are defined, suboption 04 is ignored.

For example:

```
option NEW_OP_V6.alt-image-file-name "ZTP_IMAGES/jinstall-qfx-5-20.2-alternate-img.tgz";
```

- Suboption 05: The port that the device uses to download either the image or configuration file or both instead of the default port.

```
option NEW_OP_V6.port-number 8080;
```

- Suboption 06: The JLoader package file name (supported only on QFX5100 devices)

```
option NEW_OP_V6. "jloader.tgz";
```

- Suboption 07: FTP timeout code.

```
option NEW_OP_V6. ftp-timeout "val";
```

- Suboption 08: HTTP proxy server information that is passed from the DHCP server to the DHCP client. This is useful when a device needs to access the phone-home server or redirect server via a proxy server.



NOTE: When you configure the DHCP server and HTTP proxy server, make sure that you use the correct port number to allow traffic to flow through the secure tunnel. Also, make sure that the hostname or IP address of the HTTP proxy server and port number are separated by a colon: for example, "http://[2001::1]:3128. If you don't use a colon, port 1080 is used.

When the DHCP client receives the HTTP proxy server information, it is saved in the `/var/etc/phc_v6_vendor_specific_info.xml` (INET6) file.

You can renew the HTTP proxy server information by issuing the `request dhcp client renew interface` command. The DHCP client fetches the valid HTTP proxy server information from the DHCP server. Using the command is simpler than having to restart the provisioning process. When the HTTP proxy server is renewed, or the HTTP proxy server information is changed or deleted, `jdhcp` will rewrite the `/var/etc/phc_v6_vendor_specific_info.xml` file with the latest information received from suboption 8.

```
option dhcp6.vendor-opts code 17 = string;
option NEW_OP.proxyv6-info code 8 = text;
```

- The DHCPv6 protocol defines the Vendor-specific Information Option ("VSIO") in order to send vendor options encapsulated in a standard DHCP option.

```
option vsio.NEW_OP_V6 code 2636 = encapsulate NEW_OP_V6;
```

The following example configuration shows the DHCPv6 options you've just configured:

```
subnet6 2001:db8::/32 {
    range6 2001:db8::10 2001:db8::40;
}
host chocolate {
    option host-name chocolate;
    hardware ethernet AA:BB:CC:DD:EE:FF;
    fixed-address6 2001:db8::11;
    option dhcp6.bootfile-url "http://ztp:welcome@[2001:db8::1]";

    option NEW_OP_V6.image-file-name "ZTP_IMAGES/jinstall-qfx-5-20.2I-img.tgz";
    option NEW_OP_V6.port-number 8080;
    option NEW_OP_V6.config-file-name "ZTP_FILES/baseline_config";
    option NEW_OP_V6.image-file-type symlink;
    option NEW_OP_V6.transfer-mode "https";
    option NEW_OP_V6.jloader-file "jloader.tgz ";
    option dhcp6.vendor-opts code 17 = string;
    option NEW_OP.proxyv6-info "http://[2001::1]:3128";
}
```

8. Power on the device with the default configuration.
9. Monitor the ZTP process by looking at the console.



NOTE: When SLAX scripts are executed, the `op-script.log` and `event-script.log` files are produced.

You can also use these log files to troubleshoot in case something goes wrong.

- `/var/log/dhcp_logfile`

Use this file to check DHCP client logs.

- `/var/log/event-script.log`

Use this file to check configuration commit status.

- /var/log/image_load_log

Use this file to check software image and configuration file fetch and installation status.

- /var/log/messages

Use this file to check system-level logs.

- /var/log/op-script.log

Use this file to check configuration commit status.

- /var/log/script_output

Use this file to check script execution output.

You can also monitor the ZTP process by looking at error messages and issuing operational commands. See ["Monitoring Zero Touch Provisioning" on page 506](#) for more information.

Zero Touch Provisioning on SRX Series Firewalls

IN THIS SECTION

- [Understanding Zero Touch Provisioning on SRX Series Firewalls | 496](#)
- [Configuring Zero-Touch Provisioning on an SRX Series Firewall | 501](#)
- [Understanding Factory-Default Configuration on SRX Series Firewall for Zero Touch Provisioning | 505](#)

Understanding Zero Touch Provisioning on SRX Series Firewalls

IN THIS SECTION

- [Understanding ZTP on SRX Series Firewalls | 497](#)
- [Network Activator Overview | 498](#)
- [Limitations | 501](#)

This topic includes following sections:

Understanding ZTP on SRX Series Firewalls

Zero Touch Provisioning (ZTP) enables you to provision and configure devices automatically, minimizing most of the manual intervention required for adding devices to a network. ZTP is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.

Starting in Junos OS Release 20.2R1 on SRX300, SRX320, SRX340, SRX345, SRX550 HM, and SRX1500 devices, you can use Zero Touch Provisioning with DHCP options to provision your device. See ["Zero Touch Provisioning Using DHCP Options" on page 468](#) for more information.



NOTE: ZTP is currently supported on xe-0/0/16 and xe-0/0/17 ports on SRX1500. To avoid using these ports, the following the default configurations have been added to xe-0/0/18 and xe-0/0/19 ports.

```
set interfaces xe-0/0/18 unit 0 family inet dhcp vendor-id Juniper-srx1500
set interfaces xe-0/0/19 unit 0 family inet dhcp vendor-id Juniper-srx1500
set security zones security-zone untrust interfaces xe-0/0/18.0 host-inbound-traffic
system-services dhcp
set security zones security-zone untrust interfaces xe-0/0/18.0 host-inbound-traffic
system-services tftp
set security zones security-zone untrust interfaces xe-0/0/18.0 host-inbound-traffic
system-services https
set security zones security-zone untrust interfaces xe-0/0/19.0 host-inbound-traffic
system-services dhcp
set security zones security-zone untrust interfaces xe-0/0/19.0 host-inbound-traffic
system-services tftp
set security zones security-zone untrust interfaces xe-0/0/19.0 host-inbound-traffic
system-services https
```

ZTP on SRX Series Firewalls is responsible for the initial bootup and configuration of the device when the device is powered on. This functionality includes:

- Providing the bare-minimum bootstrapping of the device. The SRX Series Firewall is shipped with a factory-default configuration. The factory-default configuration includes the URL of the redirect server, that is used to connect to the central server by using a secure encrypted connection.
- Automatically connecting to the server over the Internet, and downloading the configuration and Junos OS image as specified by the customer or user from the server when the SRX Series Firewall boots up with the factory-default configuration. The new image is installed first and then the initial configuration is applied and committed on the SRX Series Firewall.

ZTP offers the following advantages:

- Simplified and faster deployment

- Increased configuration accuracy
- Support for scaling of network without additional resources

The ZTP process uses Network Activator to initially provision SRX Series Firewalls.

Network Activator Overview

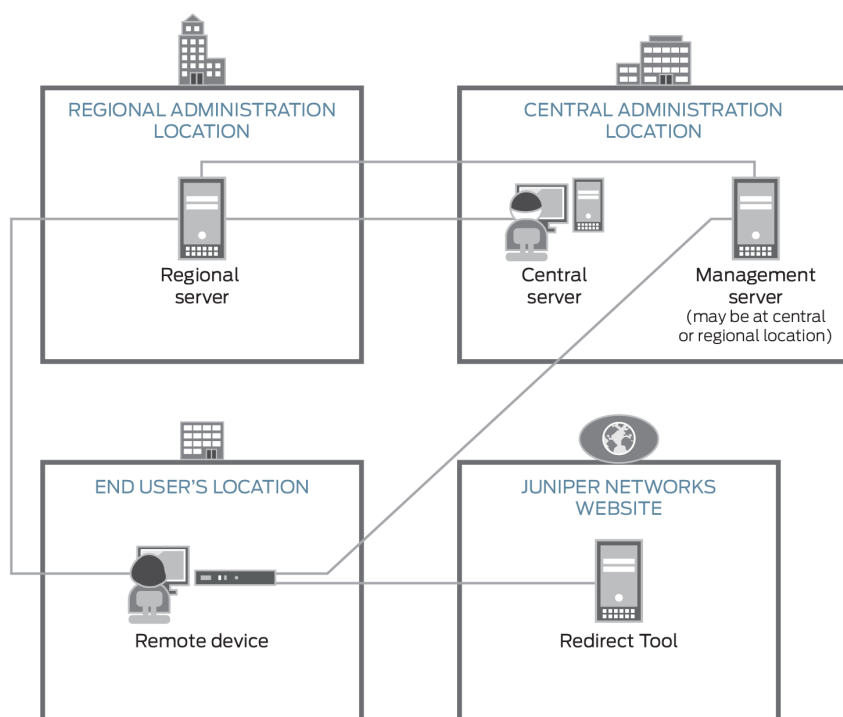
Network Service Activator enables fast device discovery and provisioning for automated configuration to eliminate complex device setup.

Network Activator initially provisions SRX Series Firewalls (henceforth referred to as *remote devices* in this documentation), which reside at end users' sites. The remote devices download a boot image and initial configuration files from servers hosting Network Activator, using a process that provides full authorization and authentication for all interactions. When initial provisioning is complete, the remote device communicates with a management server, which then starts to manage and monitor the remote device.

Network Activator uses a distributed architecture to support remote devices. Network Activator is installed on one central administration server (central server) and multiple regional administration servers (regional servers). A device communicates directly with its assigned regional server. The distributed architecture optimizes the efficiency of the initial provisioning process, contributing to high performance and scaling of the network.

[Figure 19 on page 499](#) illustrates the distributed architecture and the components involved in the initial provisioning process.

Figure 19: Components Involved in Initial Provisioning of Remote Device

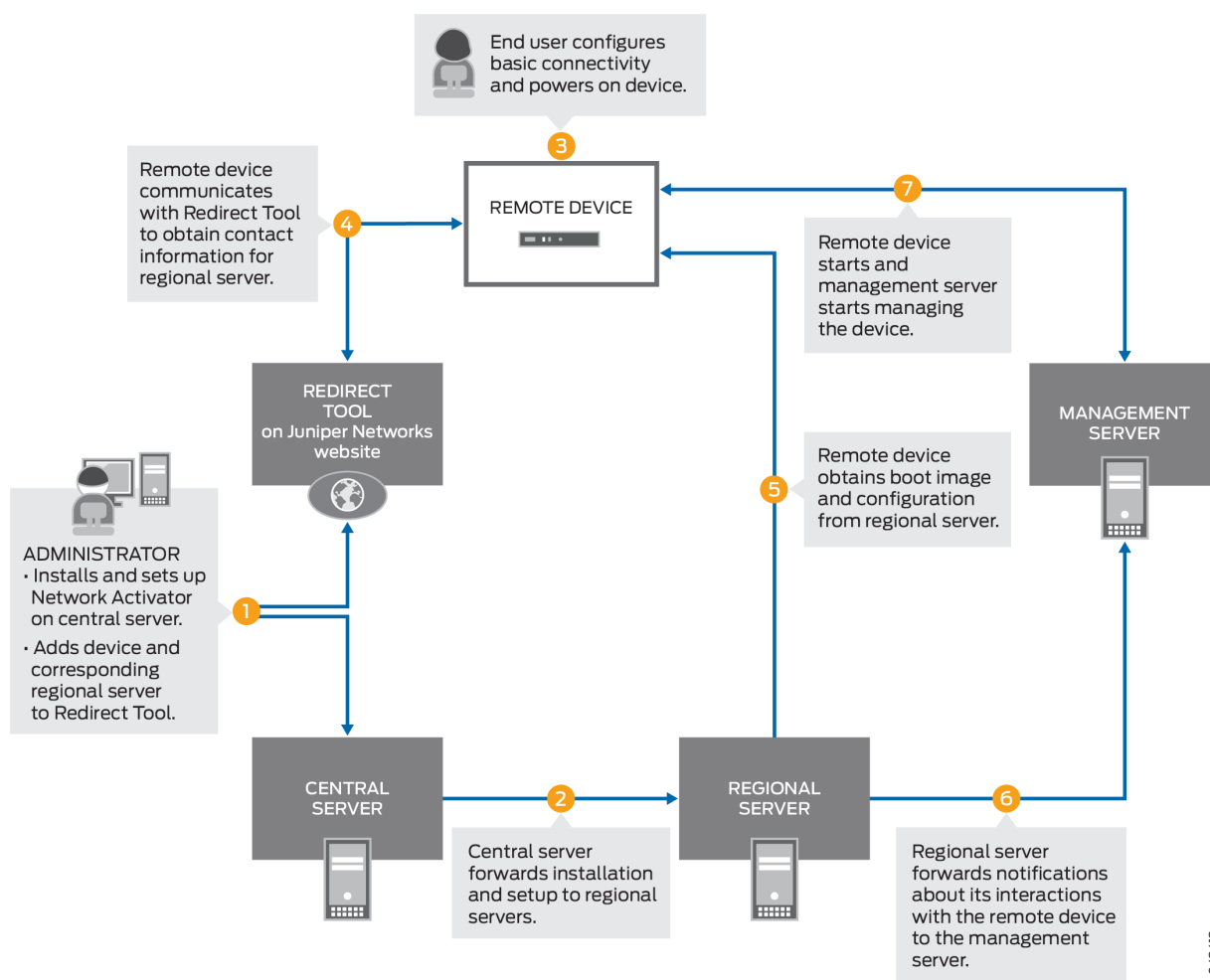


The roles of the components in the initial provisioning process are as follows:

- The remote device sends requests for initial provisioning. The remote device resides at the end user's location.
- The Redirect Tool provides authentication and authorization for remote devices to access their assigned regional servers through use of ITU-T X.509 private key infrastructure (PKI) digital certificates. Redirect service is hosted on Amazon Web Services (AWS), operated and maintained by Juniper Networks.
- The central server hosts Network Activator and communicates with the regional activator servers. Administrators at a service provider or central enterprise location interact with this server to install and set up Network Activator. The central server is located at a central geographic location for the service provider.
- The regional server also hosts Network Activator. This server stores information about its assigned remote devices and communicates directly with those devices. This server typically resides at a regional administrative location the provider designates for the end user.

Figure 20 on page 500 illustrates the initial provisioning workflow.

Figure 20: Workflow for Initial Provisioning



In detail, the provisioning workflow proceeds as follows:

1. The administrator at the service provider:
 - Installs and sets up Network Activator on the central server.
 - Adds remote devices and regional servers in the Redirect Tool.
2. The central server forwards the installation to the regional servers.
3. The end user powers on the remote device, connects it to a computer, and enters the authentication code in the webpage to send a request for initial provisioning.
4. The device transmits its X.509 certificate and fully qualified domain name (FQDN) as a provisioning request to the Redirect Tool.

5. The Redirect Tool searches its data store for the regional server that the administrator specified for this device, and confirms that the device's request corresponds to the X.509 certificate specified for the server.
6. The Redirect Tool sends contact information for the regional server to the device.
7. The device sends a request to the regional server for the URL of the boot image and the location of the initial configuration.
8. The regional server sends the information to the device.
9. The device obtains the boot image and configuration from the regional server.
10. The device uses the boot image and configuration to start and become operational.

Limitations

- There are no restrictions on the number of attempts for entering the correct activation code.
- If the remote device is not able to reach the server (because the configured address in the factory-default configuration is not correct or the server is down, and so on), the remote device attempts to connect to an alternative server (if configured in the factory-default configuration). If there is only one server configured, then you can reattempt to connect. In such scenarios, we recommend that you configure the device manually through the console.
- Captive portal redirection, required for automatically redirecting users to the authentication webpage for entering the activation code, is not supported. You must manually navigate to the activation page after connecting to the device.

Configuring Zero-Touch Provisioning on an SRX Series Firewall

Before you begin:

- Unpack the device, install it, complete the necessary cabling, connect a laptop or any other terminal device, and power on the device. See the *Hardware installation Guide* for your device for more information.
- For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, connect the management device and access the J-Web interface.

For more information, see Quick Start guides of respective devices at [SRX300](#), [SRX320](#), [SRX340](#), [SRX345](#), and [SRX550M](#).

You are provided with an option to use ZTP; you can use this option or skip it and continue with J-Web wizards.

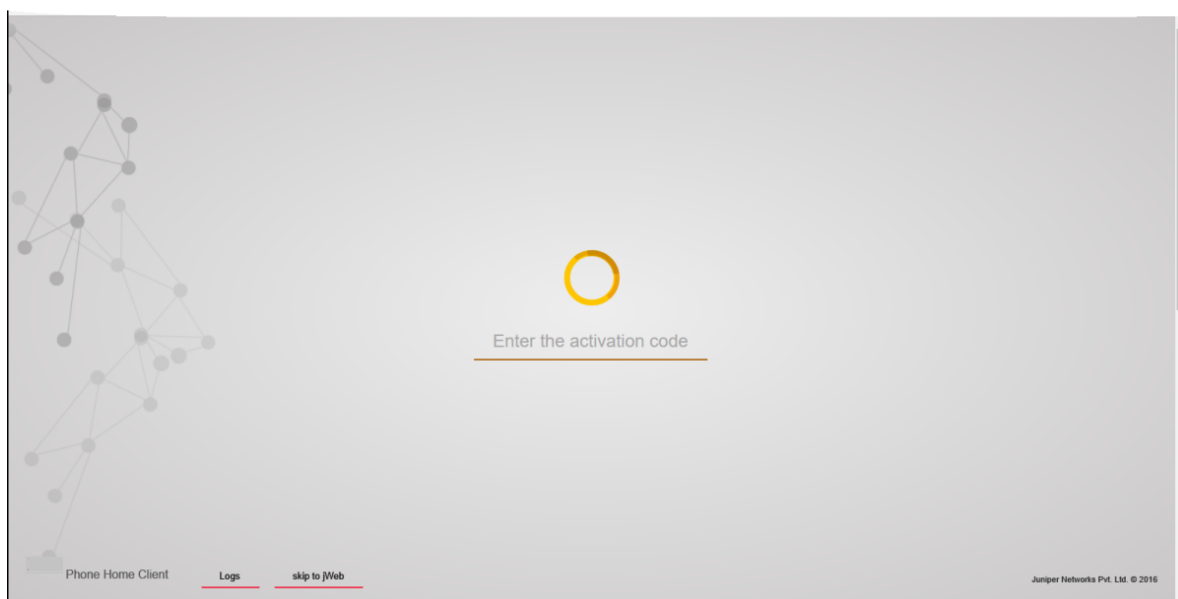
- For SRX1500 devices, before you can use J-Web to configure your device, you must access the CLI to configure the root authentication and the management interface. For more information, see [How to Set Up Your SRX1500 Services Gateway](#).

This section provides step-by-step instructions on how to use ZTP on an SRX Series Firewall for initial provisioning of the device.

To provision an SRX Series Firewall by using ZTP:

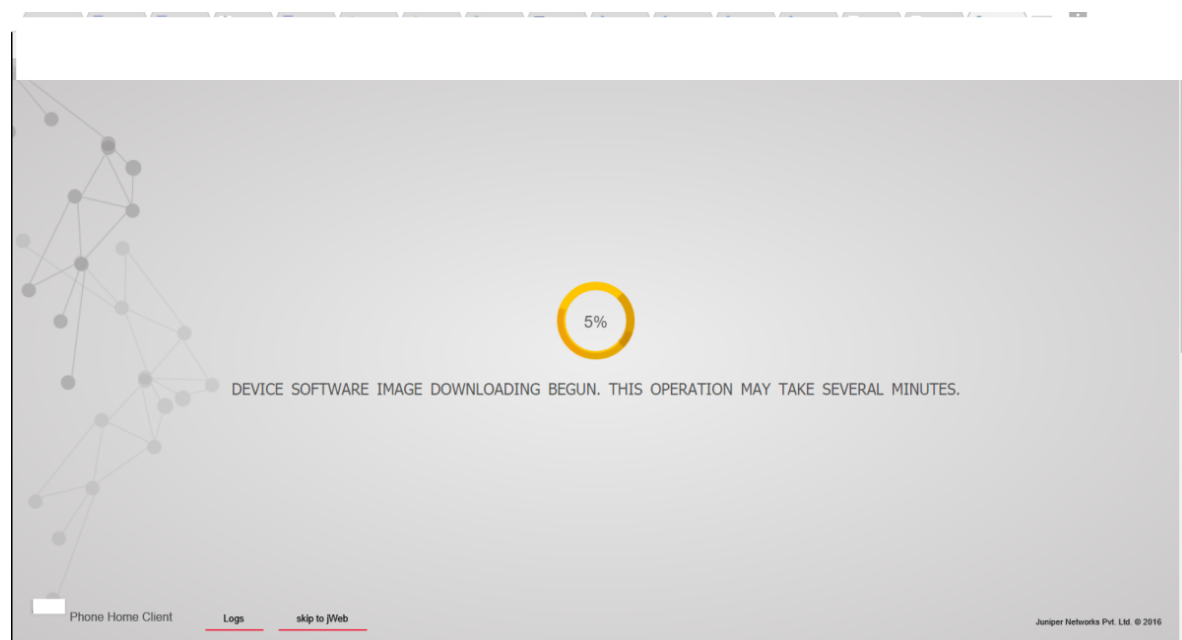
1. Connect a management device (PC or laptop) to any front panel Ethernet port (WAN port) of the SRX Series Firewall.
2. Launch a Web browser from the management device and enter the authentication code in the webpage as shown in [Figure 21 on page 502](#).

Figure 21: Entering Activation Code for ZTP



After the device is successfully authenticated, it starts downloading the software image and initial configuration from the server as shown in [Figure 22 on page 503](#).

Figure 22: Initiating ZTP Process (Software Image Downloading)

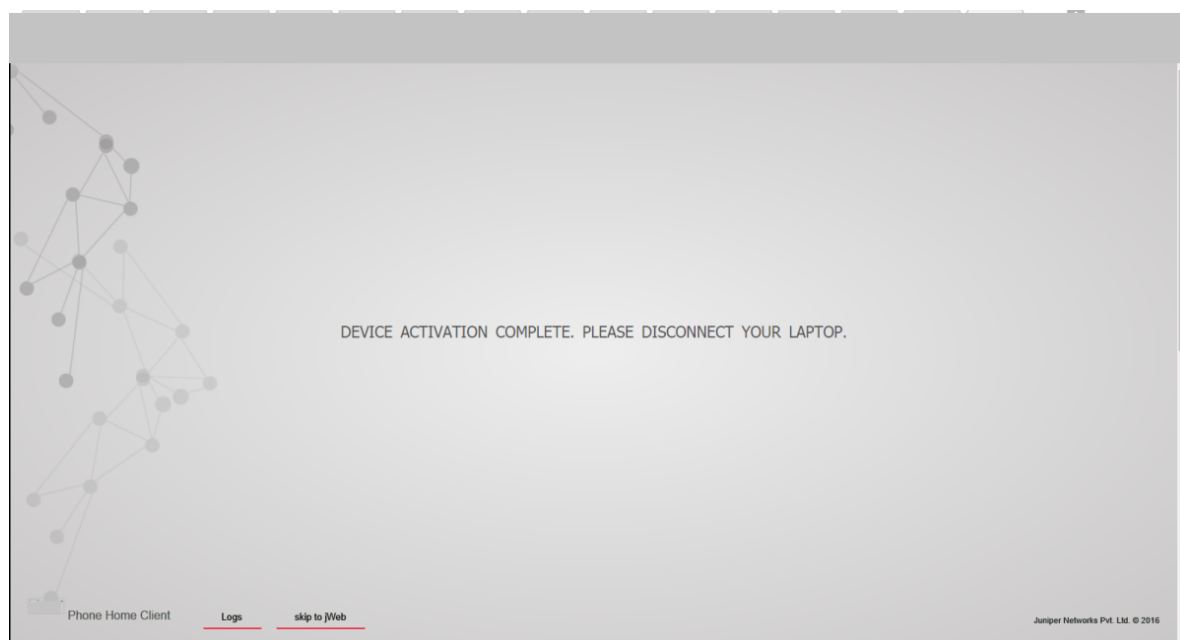


At this step:

- The activation code is sent to the server, and if the authentication is successful, the server pushes the initial configuration to the device. If the authentication is unsuccessful, you are asked to provide the correct code.
- The server can optionally push a new software image on the SRX Series Firewall. In that case, the new image is installed first and then the initial configuration is applied and committed on the device.

The new image is installed and then the initial configuration is applied and committed on the device. When the process is complete, a confirmation message is displayed, as shown in [Figure 23 on page 504](#).

Figure 23: Completing ZTP Process



3. Click **Logs** to display details of the bootstrapping process.

After successfully installing the new software image and configuration on the system, the client sends the bootstrap-complete notification to the server that provided the image and the configuration. After the notification is sent, the configuration that includes the names of servers is deleted from the system. When you use ZTP the next time, you must explicitly configure the URL of the redirect server.



NOTE: In case of failure at any stage, the procedure is started all over again.

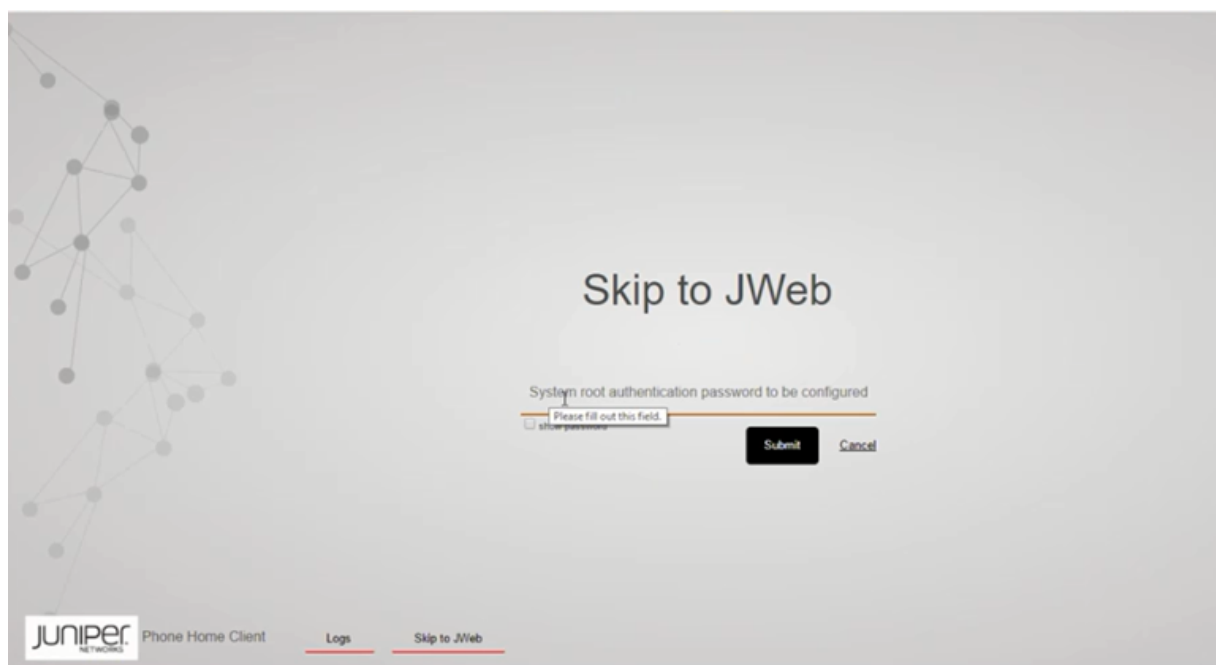


NOTE: The ZTP process either upgrades or downgrades the Junos OS version. During a downgrade on an SRX Series Firewall, if you downgrade to a software version earlier than Junos OS Release 15.1X49-D100, in which ZTP is not supported, the autoinstallation phase of the ZTP process does not happen.

For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, ZTP is the default method for provisioning the devices. However, if you want to use J-Web-based provisioning (J-Web setup wizards supported for the SRX300 line of devices and SRX550M devices), then instead of ZTP, you can use the option provided in the client portal to skip to J-Web setup wizards for performing the initial software configuration of your device.

If you select the **Skip to JWeb** option, you must configure the system root authentication password as shown in [Figure 24 on page 505](#).

Figure 24: Configuring System Root-Authentication Password



NOTE: For SRX1500 devices, the **Skip to JWeb** option is not supported. To access J-Web, the ZTP client configuration must be deleted during the initial setup of SRX1500 through CLI.

Understanding Factory-Default Configuration on SRX Series Firewall for Zero Touch Provisioning

Your services gateway is shipped with a factory-default configuration. Following is a sample of the default configuration that includes configuration for ZTP:

```
system {
  phone-home {
    rfc-compliant;
    server https://redirect.juniper.net;
  }
}
```

Note that, in this configuration:

- **server** indicates the name or IP address of the server. The factory-default configuration on an SRX Series Firewall might include IP addresses of more than one servers.

- `rfc-compliant` indicates that after an upgrade, the server enforces certain behaviors that are compliant with RFC standards.



NOTE: By default, the system autoinstallation configuration is part of the factory-default configuration of the device. So, the administrator must ensure that the configuration file sent from the regional server to the remote device (SRX Series Firewall) must include the `delete system autoinstallation` option in the factory-default configuration.

Monitoring Zero Touch Provisioning

IN THIS SECTION

- [Using the Console to Monitor Zero Touch Provisioning in Junos OS | 506](#)
- [Using System Log Alerts to Monitor Zero Touch Provisioning | 507](#)
- [Using Error Messages to Monitor Zero Touch Provisioning | 508](#)
- [Using System Log Files to Monitor Zero Touch Provisioning in Junos OS Using DHCP Options | 508](#)
- [Using System Log Files to Monitor Zero Touch Provisioning in Junos OS Using DHCPv6 Options | 510](#)
- [Using the `show dhcp client binding` Command | 511](#)
- [Using the `show dhcpv6 client binding` Command | 512](#)
- [Using the `show dhcp client statistics` Command | 513](#)
- [Using the `show dhcpv6 client statistics` Command | 514](#)

You can use the console and operational mode commands to monitor Zero Touch Provisioning.

Using the Console to Monitor Zero Touch Provisioning in Junos OS

The following Zero Touch Provisioning (ZTP) activities are displayed on the console during the ZTP process:

- Starting and ending times of ZTP process.
- Lists of bound and unbound DHCP client interfaces.
- DHCP options that DHCP servers send to DHCP clients.
- Logs indicating which interfaces are used for ZTP.

- ZTP parameters that DHCP clients obtain from DHCP servers.
- Filenames of configuration and image files, names of file servers, protocols used to fetch files, and times when DHCP servers fetch configuration and image files.
- Failure states caused by files not being on servers, or unreachable servers, and time outs.
- Number of attempts made, and number of attempts remaining, for retry in current ZTP cycle.
- Completion of file transfers.
- Installation, reboot, and state of ZTP process.
- Internal state errors and termination of ZTP process.
- Logs for when default routes were added or deleted.

Using System Log Alerts to Monitor Zero Touch Provisioning

IN THIS SECTION

- [Purpose | 507](#)
- [Action | 507](#)
- [Meaning | 508](#)

Purpose

In this example, the system log alert alerts you that the auto-image upgrade will start.

Action

Use the following system log alert to monitor the auto-image upgrade process.

```
"ALERT:Auto-image upgrade will start. This can terminate config CLI session(s). Modified
configuration will be lost. To stop Auto-image, in CLI do the
following: 'edit; delete chassis auto-image-upgrade; commit'."
```

```
"Checking whether image upgrade is already invoked"
```

Meaning

This system log alert indicates that the auto-image upgrade will start, and provides information on how to stop the auto-image upgrade process.

Using Error Messages to Monitor Zero Touch Provisioning

IN THIS SECTION

- [Purpose | 508](#)
- [Action | 508](#)
- [Meaning | 508](#)

Purpose

Error messages provide information on which DHCP options are not configured.

Action

Use the information in the following error message to find out which DHCP options are not configured.

```
"DHCP Log Server Option"  
"DHCP Host Name Option"  
"DHCP NTP Server Option"
```

Meaning

The error message indicates that the DHCP log server, hostname, and NTP server options are not configured.

Using System Log Files to Monitor Zero Touch Provisioning in Junos OS Using DHCP Options

IN THIS SECTION

- [Purpose | 509](#)

- Action | 509
- Meaning | 509

Purpose

System log files provide information on the state of the auto-upgrade process, lists of bound and unbound DHCP client interfaces, IP addresses of file servers, names and locations of image and configuration files, and successful and failed attempts at fetching configuration and image files.

Action

Use the information in the following system log files to monitor the auto-upgrade process.

```
Auto Image Upgrade: Start fetching config-file file from server 10.1.1.1 through irb using ftp
```

```
Auto Image Upgrade: Tried [2] attempts to fetch config-file file from server 10.1.1.1 through  
irb. Summary: "Retrieving /config-file  
:: Failed to open file.". To retry [4] times.
```

```
Auto Image Upgrade: Tried [4] attempts to fetch config-file file from server 10.1.1.1 through  
irb. Summary: "Retrieving /config-fileconfig-file  
:: Failed to open file.". To retry [2] times.
```

```
Auto Image Upgrade: Tried [6] attempts to fetch config-file file from server 10.1.1.1 through  
irb. Summary: "Retrieving /config-file  
:: Failed to open file.". To retry [0] times.
```

```
Auto Image Upgrade: All [6] attempts to fetch config-file file from server 10.1.1.1 through irb  
FAILED. Start retry again in few minutes.
```

Meaning

These system log files indicate that there were six failed attempts to fetch the configuration file from the file server, the IP address of the file server, the DHCP client interface name, and the number of times the retry process occurred.

Using System Log Files to Monitor Zero Touch Provisioning in Junos OS Using DHCPv6 Options

IN THIS SECTION

- Purpose | 510
- Action | 510
- Meaning | 511

Purpose

System log files provide information on the state of the auto-upgrade process, lists of bound and unbound DHCP client interfaces, IP addresses of file servers, names and locations of image and configuration files, and successful and failed attempts at fetching configuration and image files.

Action

Use the information in the following system log files to monitor the auto-upgrade process.

```
Auto Image Upgrade: Tried [2] attempts to fetch junos-vmhost-install-20.2.tgz file from server 2001:db8::1 through et-0 /0/0:2. Summary: "fetch-secure: https://[2001*: Connection refused". To retry [4] times.
```

```
Auto Image Upgrade: Tried [4] attempts to fetch junos-vmhost-install-20.2.tgz file from server 2001:db8::1 through et-0 /0/0:2. Summary: "fetch-secure: https://[2001*: Connection refused". To retry [2] times.
```

```
Auto Image Upgrade: Tried [6] attempts to fetch junos-vmhost-install-20.2.tgz file from server 2001:db8::1 through et-0 /0/0:2. Summary: "fetch-secure: https://[2001*: Connection refused". To retry [0] times.
```

Meaning

These system log files indicate that there were six failed attempts to fetch the image file from the file server, the IP address of the file server, the DHCPv6 client interface name, and the number of times the retry process occurred.

Using the show dhcp client binding Command

IN THIS SECTION

- Purpose | 511
- Action | 511
- Meaning | 511

Purpose

Issue the show dhcp client binding command to display DHCP client binding information

Action

Issue the show dhcp client binding command to display the IP address of the DHCP client, the hardware address of the DHCP client, number of seconds in which the DHCP client's IP address lease expires, state of the DHCP client IP address in the binding table, and the name of the interface that has active client bindings.

show dhcp client binding

```
user@device# show dhcp client binding
```

IP address	Hardware address	Expires	State	Interface
10.0.0.0	00:22:83:2a:db:dc	0	SELECTING	irb.0
10.6.6.13	00:22:83:2a:db:dd	49201	BOUND	vme.0
10.0.0.0	00:22:83:2a:db:df	0	SELECTING	xe-0/0/0.0
10.0.0.0	00:22:83:2a:db:e0	0	SELECTING	xe-0/0/1.0

Meaning

The output of this command shows that there is one client interface that is bound, and that there are three interfaces that are receiving DHCP offers from the DHCP server.

Using the show dhcpv6 client binding Command

IN THIS SECTION

- Purpose | 512
- Action | 512
- Meaning | 512

Purpose

Issue the show dhcpv6 client binding command to display DHCP client binding information

Action

Issue the show dhcpv6 client binding command to display the IP address of the DHCPv6 client, the hardware address of the DHCPv6 client, number of seconds in which the DHCPv6 client's IP address lease expires, state of the DHCPv6 client IP address in the binding table, and the name of the interface that has active client bindings.

show dhcpv6 client binding

```
user@device# show dhcpv6 client binding
```

IP/prefix	Expires	State	ClientType	Interface	Client
DUID					
2001:db8::10		57	SELECTING	STATEFUL	em0.0
LL0x3-54:4b:8c:d3:a2:34					
2001:db8::10		46	SELECTING	STATEFUL	em2.0
LL0x3-54:4b:8c:d3:a2:35					
2001:db8::10		38	SELECTING	STATEFUL	et-0/0/0:0.0
LL0x3-54:4b:8c:d3:a2:3b					
2001:db8::10		530	BOUND	STATEFUL	et-0/0/0:1.0
LL0x3-54:4b:8c:d3:a2:3c					

Meaning

The output of this command shows that there is one client interface that is bound, and that there are three interfaces that are receiving DHCPv6 offers from the DHCP server.

Using the show dhcp client statistics Command

IN THIS SECTION

- Purpose | 513
- Action | 513
- Meaning | 514

Purpose

Issue the show dhcp client statistics command to display DHCP client statistics.

Action

Issue the show dhcp client statistics command to display DHCP client statistics, such as the number of packets dropped, and the number DHCP and BOOTP messages sent and received.

show dhcp client statistics

```
user@device# show dhcp client statistics
Packets dropped:
  Total                14
  Send error           14
Messages received:
  BOOTREPLY            5
  DHCPOFFER             1
  DHCPACK               4
  DHCPNAK               0
  DHCPFORCERENEW       0
Messages sent:
  BOOTREQUEST          6751
  DHCPDECLINE           0
  DHCPDISCOVER         6747
  DHCPREQUEST           4
  DHCPINFORM            0
  DHCPRELEASE           0
  DHCPRENEW             0
  DHCPREBIND            0
```

Meaning

The output of this command displays how many packets were dropped with errors, the number of BOOTREPLY and DHCPOFFER messages that were received, and the number of BOOTREQUEST and DHCPREQUEST messages that were sent.

Using the show dhcpv6 client statistics Command

IN THIS SECTION

● Purpose | 514

● Action | 514

● Meaning | 515

Purpose

Issue the show dhcpv6 client statistics command to display DHCPv6 client statistics.

Action

Issue the show dhcpv6 client statistics command to display DHCPv6 client statistics, such as the number of packets dropped, and the number of DHCPv6 messages sent and received.

show dhcpv6 client statistics

```

user@device# show dhcpv6 client statistics
Dhcpv6 Packets dropped:
  Total                20323
  Bad Send              7580
  Bad Options           12743

Messages received:
  DHCPV6_ADVERTISE      13
  DHCPV6_REPLY          109
  DHCPV6_RECONFIGURE     0

Messages sent:
  DHCPV6_DECLINE         0
  
```

DHCPV6_SOLICIT	879
DHCPV6_INFORMATION_REQUEST	0
DHCPV6_RELEASE	0
DHCPV6_REQUEST	9
DHCPV6_CONFIRM	0
DHCPV6_RENEW	61
DHCPV6_REBIND	41

Meaning

The output of this command displays how many packets were dropped with errors, and the number of DHCPV6 messages that were received and sent.

Change History Table

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

Release	Description
21.4R1-EVO	Starting in Junos OS Evolved Release 21.4R1 on the QFX5130-32CD, QFX5220, and QFX5700 devices, ZTP supports the DHCPv6 client on the management interface. During the bootstrap process, the device first uses the DHCPv4 client to request for information regarding image and configuration file from the DHCP server. The device checks the DHCPv4 bindings sequentially. If there is a failure with one of the DHCPv4 bindings, the device will continue to check for bindings until provisioning is successful. If there are no DHCPv4 bindings, however, the device will check for DHCPv6 bindings and follow the same process as for DHCPv4 until the device can be provisioned successfully. The DHCP server uses DHCPv6 options 59 and 17 and applicable sub-options to exchange ZTP-related information between itself and the DHCP client.
21.3R1-EVO	Starting in Junos OS Evolved Release 21.3R1, on PTX10001-36MR, PTX10003, PTX10004, PTX10008, and PTX10016 devices, ZTP now supports DHCP options 61 and 77. DHCP option 61 is used to specify the chassis serial number, and DHCP option 77 is used to specify the make, model, and software version of the chassis.
21.2R1-EVO	Starting in Junos OS Evolved Release 21.2R1 on PTX10008 devices, Zero Touch Provisioning (ZTP) dynamically detects the port speed of WAN interfaces and uses this information to create ZTP server ports with the same speed.
21.2R1-EVO	Starting in Junos OS Evolved Release 21.2R1, QFX5700 devices support the ability for either WAN interfaces or management interfaces to automatically download and install the appropriate software and the configuration file on your device during the ZTP bootstrap process.

21.2R1	Starting in Junos OS Release 21.2R1 on QFX10002 devices, Zero Touch Provisioning (ZTP) dynamically detects the port speed of WAN interfaces and uses this information to create ZTP server ports with the same speed.
21.2R1	Starting in Junos OS Release 21.2R1, on EX2300-C, EX2300-MP, EX4300, EX4300-MP, EX4300-VC, EX4400-24MP, EX4400-48MP, EX4600-VC, EX4650, and EX4650-48Y-VC devices, during the bootstrapping process, the phone-home client can access the redirect server through a proxy server. The DHCP server uses DHCP option 43 suboption 8 to deliver the details of IPv4 and/or IPv6 proxy servers to the phone-home client. The DHCP daemon running on the target switch learns about the proxy servers in the initial DHCP cycle and then populates either the <code>phc_vendor_specific_info.xml</code> or the <code>phc_v6_vendor-specific_info.xml</code> files located in the <code>/var/etc/</code> directory with the vendor-specific information.
21.2R1	Starting in Junos OS Release 21.2R1, on EX2300-C, EX2300-MP, EX4300, EX4300-MP, EX4300-VC, EX4400-24MP, EX4400-48MP, EX4600-VC, EX4650, and EX4650-48Y-VC devices, you can use a DHCPv6 client and ZTP to provision a switch. During the bootstrap process, the device first uses the DHCPv4 client to request for information regarding the image and configuration file from the DHCP server. The device checks the DHCPv4 bindings sequentially. If there is a failure with one of the DHCPv4 bindings, the device continues to check for bindings until provisioning is successful. However, if there are no DHCPv4 bindings, the device checks for DHCPv6 bindings and follows the same process as for DHCPv4 until the device is provisioned successfully. Both DHCPv4 and DHCPv6 clients are included as part of the default configuration on the device. The DHCP server uses DHCPv6 options 59 and 17 and applicable suboptions to exchange ZTP-related information between itself and the DHCP client.
21.1R1	Starting in Junos OS Release 21.1R1, on EX2300, EX2300-VC, EX3400, EX3400-VC, EX4400-24T, EX4400-48F, EX4400-48T, and EX4600 devices, when the phone-home client receives information regarding the HTTP proxy server via DHCP option 43 suboption 8, it will create an HTTPS transparent tunnel with the proxy server. Once the tunnel is established, the phone-home client uses the tunnel as a proxy for the phone-home server or redirect server. The phone-home client downloads the software image and configuration file through the tunnel onto the device. Once bootstrapping is complete, the device reboots and the tunnel quits.
21.1R1	Starting in Junos OS Release 21.1R1, on EX2300, EX2300-VC, EX3400, EX3400-VC, EX4400-24T, EX4400-48F, EX4400-48T, and EX4600 devices, during the bootstrapping process, the phone-home client can access the redirect server through a proxy server. The DHCP server uses DHCP option 43 suboption 8 to deliver the details of IPv4 and/or IPv6 proxy servers to the phone-home client. The DHCP daemon running on the target switch learns about the proxy servers in the initial DHCP cycle and then populates either the <code>phc_vendor_specific_info.xml</code> or the <code>phc_v6_vendor-specific_info.xml</code> files located in the <code>/var/etc/</code> directory with the vendor-specific information.
20.4R1-EVO	Starting in Junos OS Evolved Release 20.4R1, PTX10004 devices support automation of the device configuration and software upgrade over the management interface of Routing Engine 0 (RE0).

20.4R1-EVO	Starting in Junos OS Evolved Release 20.4R1, ACX5448 and QFX5120-48YM devices support the ability for either WAN interfaces or management interfaces to automatically download and install the appropriate software and the configuration file on your device during the ZTP bootstrap process.
20.4R1	Starting in Junos OS Release 20.4R1 on the MX-Series, EX3400, EX4300, QFX5100, and QFX5200 devices, ZTP supports the DHCPv4 client. During the bootstrap process, the device first uses the DHCPv4 client to request for information regarding image and configuration file from the DHCP server. The device checks the DHCPv4 bindings sequentially. If there is a failure with one of the DHCPv4 bindings, the device will continue to check for bindings until provisioning is successful. If there are no DHCPv4 bindings, however, the device will check for DHCPv6 bindings and follow the same process as for DHCPv4 until the device can be provisioned successfully. The DHCP server uses DHCPv6 options 59 and 17 and applicable sub-options to exchange ZTP-related information between itself and the DHCP client.
20.4R1	Starting in Junos OS Release 20.4R1 on the EX4600, EX4650, EX9200 with RE-S-EX9200-2X00X6, QFX5110, QFX5200, QFX5210, QFX5120-32C, and QFX5120-48Y devices, you can use either the legacy DHCP-options-based ZTP or the phone-home client (PHC) to provision software for the switch. When the switch boots up, if there are DHCP options that have been received from the DHCP server for ZTP, ZTP resumes. If DHCP options are not present, PHC is attempted. PHC enables the switch to securely obtain bootstrapping data, such as a configuration or software image, with no user intervention other than having to physically connect the switch to the network. When the switch first boots up, PHC connects to a redirect server, which redirects to a phone home server to obtain the configuration or software image.
20.2R1-S1	Starting in Junos OS Release 20.2R1-S1 on the MX-Series, EX3400, EX4300, QFX5100, and QFX5200 devices, ZTP supports the DHCPv6 client. During the bootstrap process, the device first uses the DHCPv4 client to request for information regarding image and configuration file from the DHCP server. The device checks the DHCPv4 bindings sequentially. If there is a failure with one of the DHCPv4 bindings, the device will continue to check for bindings until provisioning is successful. If there are no DHCPv4 bindings, however, the device will check for DHCPv6 bindings and follow the same process as for DHCPv4 until the device can be provisioned successfully. The DHCP server uses DHCPv6 options 59 and 17 and applicable sub-options to exchange ZTP-related information between itself and the DHCP client.
20.2R1	Starting in Junos OS Release 20.2R1 on SRX300, SRX320, SRX340, SRX345, SRX550 HM, and SRX1500 devices, you can use Zero Touch Provisioning with DHCP options or the phone-home client to provision your device.
20.1R1-EVO	Starting in Junos OS Evolved Release 20.1R1 on PTX10003 devices, Zero Touch Provisioning (ZTP) dynamically detects the port speed of WAN interfaces and uses this information to create ZTP server ports with the same speed.

20.1R1-EVO	Starting in Junos OS Evolved Release 20.1R1, PTX10008 devices support automation of the device configuration and software upgrade over the management interface of Routing Engine 0 (RE0).
19.4R1	Starting in Junos OS Release 19.4R1, ZTP can automate the provisioning of the device configuration and software image on Juniper Route Reflector (JRR). ZTP supports self image upgrades and automatic configuration updates using ZTP DHCP options. In this release, ZTP supports revenue ports em2 thru em9, in addition to management port em0 which is supported in Junos OS Releases before 19.4R1.
19.3R1-Evo	Starting in Junos OS Evolved Release 19.3R1, on QFX5220-128C device, in Zero Touch Provisioning (ZTP), you can use either WAN interfaces or management interfaces, to automatically download and install the appropriate software and the configuration file on your device during the bootstrap process.
19.3R1	Starting in Junos OS Release 19.3R1, you can use either WAN interfaces or management interfaces to automatically download and install the appropriate software and the configuration file on your router during the ZTP bootstrap process.
19.2R1	Starting in Junos OS Release 19.2R1, ZTP can automate the provisioning of the device configuration and software image on management interface em0 for ACX5448 switches.
19.1R1-EVO	Starting in Junos OS Evolved Release 19.1R1, ZTP can automate the provisioning of the device configuration and software image on the management interface for QFX5220 and PTX10003 devices.
19.1-Evo	Starting in Junos OS Evolved Release 19.1R1, to monitor zero touch provisioning on Junos OS Evolved, use the <i>show system ztp</i> command.
18.3R1	Starting in Junos OS Release 18.3R1, ZTP, which automates the provisioning of the device configuration and software image with minimal manual intervention, is supported on MX Series VM hosts.
18.2R1	Starting in Junos OS Release 18.2R1, ZTP can automate the provisioning of the device configuration and software image on VM host platforms that use PTX5000, PTX3000, PTX10008, PTX10016, PTX10002-60C routers.
18.2R1	Starting in Junos OS Release 18.2R1, ZTP can automate the provisioning of the device configuration and software image on VM host platforms that use QFX10008 and QFX10016 switches.
18.1R1	Starting in Junos OS Release 18.1R1, ZTP can automate the provisioning of the device configuration and software image on VM host platforms that use QFX10002-60C switches.

17.2R1	Starting in Junos OS Release 17.2R1, ZTP can automate the provisioning of the device configuration and software image on VM host platforms that use PTX1000 routers.
16.1R1	Starting in Junos OS Release 16.1R1, you can provision supported devices by using either a script to be executed or a configuration file to be loaded.
12.2	Starting in Junos OS Release 12.2, you can use the console and operational commands to monitor Zero Touch Provisioning.

Secure Zero Touch Provisioning

IN THIS SECTION

- Overview | 520
- Benefits | 520
- Use Case | 521
- SZTP Requirements | 521
- SZTP Infrastructure Components | 522
- DevID Workflow | 523
- Onboarding Information | 523
- DHCP V4 Option 143 | 525
- DHCP V6 Option 135 | 526
- SZTP Workflow | 527
- SZTP for Network Devices with Dual Routing Engines | 528



NOTE: To see which platforms support Secure Zero Touch Provisioning (SZTP), go to [Feature Explorer](#). In the **Explore Features** section of the Feature Explorer page, select **All Features**. In the **Features Grouped by Feature Family** box, select Secure ZTP. You can also type the name of the feature in the **Search for Features** edit box. See the Release History Table at the end of this topic for more details of how ZTP support has expanded.

Overview



NOTE: The phone-home client (PHC) process supports Secure Zero Touch Provisioning (SZTP).

You can use RFC-8572-based SZTP to bootstrap remotely located network devices that are in a factory-default state. SZTP enables mutual authentication between the bootstrap server and the network device before provisioning the remote network device.

To enable mutual authentication, you need a unique digital voucher and DevID (Digital Device ID or Cryptographic Digital Identity) programmed network device. The DevID is embedded inside the Trusted Platform Module (TPM) 2.0 chip on the network device. Juniper Networks issues a digital voucher to customers for each eligible network device.

We support SZTP on management and WAN interfaces.



NOTE: DHCP-based legacy ZTP is disabled. We do not support DHCP-based legacy ZTP on hardware that supports SZTP.

SZTP is compliant with RFC 8572 and requires the following infrastructure to ensure the identity and authenticity of your network devices:

- Trusted Platform Module (TPM) 2.0
- Digital Device IDs (DevIDs)
- DevID Certificates
- X.509 Pinned Domain Certificates (PDCs)
- Owner Certificates
- DevID Trust Anchors
- Vouchers

For information on how to generate vouchers, see [Generate Voucher Certificate](#).

To onboard your Juniper devices with Secure ZTP, see [Secure ZTP Quick Start Guide](#).

Benefits

- You can provision a remote network device without manual intervention.

- You can provision a network device securely from a central location, which prevents unauthorized entities from taking control of your network device.
- Your redirect and bootstrap servers verify the authenticity of your network device based on the DevID that's programmed in the network device's TPM.
- Your network device verifies the authenticity of your redirect servers and bootstrap servers, and bootstrap information, based on the devices' vouchers.

Use Case

For network devices that are shipped from the factory, you can make the network devices operational both securely and remotely without manually touching the network device. The network device needs to be able to use the Dynamic Host Configuration Protocol (DHCP) to obtain network connectivity information and connect to a remote bootstrap server.

SZTP Requirements

To deploy SZTP in your network, you need to perform the following tasks:

1. Deploy your DHCP and DNS servers.
2. Configure DHCP V4 option 143 or DHCP V6 option 136 on your DHCP server, so the DHCP server can advertise the names of your redirect and bootstrap servers.
3. Deploy your redirect and bootstrap servers.
4. Acquire DevID trust anchors from Juniper Networks.
5. Generate owner certificates for one network device or a group of network devices.
6. Generate pinned domain certificates (PDCs) for each network domain.
7. Acquire vouchers from Juniper Networks.
8. Generate redirect and bootstrap information for each network device.
9. Use the redirect and bootstrap information that the redirect and bootstrap servers provide to provision your network devices.

After you deploy SZTP in your network, and then deploy a new network device, the network device bootstraps automatically.

SZTP Infrastructure Components

Trusted Platform Module (TPM) 2.0

The TPM is a microchip that provides security-related functions. During the manufacturing process, Juniper Networks programs the TPM with a digital device ID (DevID) and an asymmetric keypair (public key and private key). The TPM locks the private key of the asymmetric pair in a tamper-proof location.

DevIDs

The DevID corresponds to the private key and protects the private key. Applications that require signing or encryption use the DevID private key.

Applications running on your network device use the DevID private key in the network device's TPM to prove the identity of the network device to a remote verifier.

DevID Certificates

Juniper Networks generates a DevID certificate (X.509 certificate) for the public key that corresponds to the DevID of the private key. The DevID certificate contains the serial number of the network device for which the DevID was created. DevID certificate is generated conforming to the IEEE 802.1AR standard.



NOTE: We support the IDevID. We do not support the LDevID.

X.509 Pinned Domain Certificates (PDCs)

Create an X.509 pinned domain certificate (PDC) for every network domain. The PDC can be either a root CA certificate or an intermediate CA certificate. Convert the PDC from distinguished encoding rules (DER) to base 64 encoding. Make sure that the PDC is a certificate authority (CA) and conforms to X.509.

Owner Certificates

The owner certificate verifies the vendor that bought or owns the network device. Generate an asymmetric key pair (public key and private key) for each network device or group of network devices. The key pair needs to use either Rivest-Shamir-Adleman (RSA) or elliptic curve cryptography (ECC). Keep the private key protected in a secure location. The Pinned Domain Certificate (PDC) should be the CA for the owner certificate.

DevID Trust Anchors

Juniper Networks provides DevID trust anchors. Install the DevID trust anchors in redirect and bootstrap servers to verify the DevID certificate that the device or client presents while it establishes a TLS session.

Voucher Certificates

To receive voucher certificates, enter the PDC and the network device's serial number in the Juniper Agile Licensing (JAL) Portal. Once you receive the voucher certificates, include them as part of the bootstrap information on your bootstrap server. The bootstrap server provides the voucher certificates to your network devices. Your network devices then use the bootstrap information to verify the trust anchors that your redirect server provides.

For step-by-step instructions on how to receive vouchers, see [Generate Voucher Certificate](#).

DevID Workflow

1. When an application requires signing or encryption that uses the DevID, the application requests a TLS session with the bootstrap server.
2. The bootstrap server sends a TLS response to the network device asking the network device to do the following:
 - Provide its DevID certificate
 - Prove that it has a private key
3. The network device signs the session data with the DevID of the private key.
4. The network device sends the digital signature and the DevID certificate to the bootstrap server.
5. The bootstrap server uses the DevID certificate to verify the digital signature.
6. The bootstrap server uses the DevID trust anchor that Juniper Networks provides to verify the DevID certificate.

Onboarding Information

In order for a network device to bootstrap itself and establish secure connections with other systems, you need to provide onboarding information. Onboarding information is data that a network device uses to bootstrap itself and connect with other systems. When a network device sends this data, the data needs to be encoded in a format that conforms to RFC 8572.

Boot Image Information

Boot image information includes the name of the OS and the OS version. We recommend that you specify "Junos" as the OS version. Make sure that you specify the correct OS version to prevent the network device from continuously downloading and installing software.

Download URI

The download URI provides the location of the boot image.

Image Verification

The image verification field includes the hash algorithm that you use to generate a secure hash for the software image and the digest value of the software image. SZTP supports SHA256. Encode the digest value as a hexadecimal string.

Configuration Handling

SZTP can either merge or replace a configuration. Create the configuration in XML and encode the configuration to Base 64 format. The configuration needs to be in Base 64 format so that the bootstrap server can include it in its bootstrap information.

Pre-configuration Scripts

SZTP supports Bourne shell scripts and Python scripts. The Bourne shell script interpreter path is `#!/bin/sh`, and the Python interpreter path is `#!/usr/bin/python`.

If the script is a Bourne script, SZTP checks the end value of the script. If the script exits with a nonzero value, the SZTP process restarts. If the script is a Python script, SZTP doesn't check the end value of the script. The output of a script could have errors even if the script ran successfully.

Here's an example of the onboarding information in XML:

```
=====
<onboarding-information>
  <boot-image>
    <os-name>Junos</os-name>
    <os-version>22.2R1</os-version>
    <download-uri>https://example.com/path/to/image/file,https://example-1.com/path/to/image/
file</download-uri>
    <image-verification>
      <hash-algorithm> </hash-algorithm>
```

```

        <hash-
value>ba:ec:cf:a5:67:82:b4:10:77:c6:67:a6:22:ab:7d:50:04:a7:8b:8f:0e:db:02:8b:f4:75:55:fb:c1:13:b
2:33</hash-value>
    </image-verification>
</boot-image>
<configuration-handling>merge</configuration-handling>
<pre-upgrade-script>base64encodedvalue</pre-upgrade-script>
<configuration>base64encodedvalue</configuration>
<post-configuration-script>base64encodedvalue</post-configuration-script>
</onboarding-information>
=====

```

Post-configuration Scripts

The pre-configuration script requirements also apply to post-configuration scripts. If any post-configuration script fails, your device rolls back to the configuration it was running before the pre-configuration script was executed. The SZTP process restarts.

DHCP V4 Option 143

Configure DHCP V4 option 143 on your DHCP server before it can provide any IP addresses to the DHCP client.

If you use an MX-Series device as a DHCP server, enable DHCP V4 Option 143.

Here is a sample configuration:

```

access {
  address-assignment {
    pool p1 {
      family inet {
        network 192.168.2.0/24;
        range r1 {
          low 192.168.2.2;
          high 192.168.2.254;
        }
        dhcp-attributes {
          maximum-lease-time 2419200;
          server-identifier 192.168.2.1;
          router {
            192.168.2.1;
          }
        }
      }
    }
  }
}

```

```

    }
  }
  option 143 hex-string 001368747470733a2f2f6578616d706c652e636f6d;
}
}
}

```

DHCP V6 Option 135

Here is a sample configuration:

```

access {
  address-assignment {
    neighbor-discovery-router-advertisement p2;
    pool p2 {
      family inet6 {
        prefix 2001:db8::/64;
        range r1 {
          low 2001:db8:::200/128;
          high 2001:db8:::299/128;
        }
        dhcp-attributes {
          dns-server {
            2001:db8:::8888;
          }
        }
        option 135 hex-string 001a68747470733a2f2f6d782d7068732d736572766572362e6e6574;
      }
    }
  }
}

```

Converting Hexadecimal Format to ASCII Text Format

This hexadecimal text string in the DHCP V6 Option 135, for example, is equal to 26 bytes in ASCII text format. In hexadecimal format, 26 is represented as 001a. Each hexadecimal number is equal to one byte, and each byte is equal to a combination of ASCII characters.

To convert the 001a68747470733a2f2f6d782d7068732d736572766572362e6e6574 hexadecimal string to ASCII characters, you need to map the hexadecimal letters and numbers to ASCII letters, numbers, and symbols.

In this example, we're mapping the URL used for DHCP Option 135. You can use the same process for the URL used in DHCP Option 143.

Here's an example URL that shows the mapping between hexadecimal format and ASCII format. You can see that each hexadecimal number is mapped to letters and symbols in ASCII format:

```
68(h) 74(t) 74(t) 70(p) 73(s) 3A(:) 2F(/) 2F(/) 61(a) 62(b) 2D(-) 63(c) 64(d) 65(e) 2D(-) 73(s)
65(e) 72(r) 76(v) 65(e) 72(r) 36(. ) 2E (n)6E 65(e) 74(t)
```

The final URL is `https://ab-cde-server.net`.

Use a hexadecimal to ASCII converter and vice versa to make sure your results are correct.

SZTP Workflow



NOTE: This topic includes only one of the permitted workflows. We support everything in the RFC 8572 standard, including Appendix-B.

If your device isn't already in a factory-default state, issue one of the following commands to bring your device into a factory-default state.

- On network devices running Junos OS, issue the `request vmhost zeroize` command.
- For network devices running Junos OS Evolved, issue the `request system zeroize` command.

When a device boots up in a factory-default state, the following events occur.

1. The DHCP client sends a request to the DHCP server to obtain the name, IP address, or host name of either the bootstrap server or customer redirect server.

Configure either DHCP option 143 for V4 or DHCP option 135 for V6. The DHCP client requests the IP address for each bootstrap or redirect server until the device completes bootstrapping.

2. The DHCP server sends the server host name of either a bootstrap or a customer redirect server to the DHCP client.
3. The phone-home client (PHC) on your device sends a bootstrap request to the server it learned from the DHCP option. If you've provided multiple servers in the DHCP option, the device tries to bootstrap with each server sequentially.

The device tries to bootstrap with any bootstrap, customer redirect, or DNS server that the PHC learns through the DHCP option. The device attempts to bootstrap to a server in round-robin fashion until the device bootstraps successfully.

4. The bootstrap server responds with signed onboarding information along with the owner certificate and ownership voucher.
5. The PHC uses the information in the owner certificate and ownership voucher to verify the signed onboarding information.
6. The PHC extracts image and configuration information.
7. If the device is running a different image, the device downloads the image, uses the new image to upgrade, and then reboots with the new image.

Post reboot, the entire SZTP sequence repeats, except that device doesn't reboot because it already has the required image.

8. The PHC commits the configuration.
9. (Optional) The PHC runs post-configuration scripts.
10. The PHC sends a bootstrap complete message to the PHS.
11. The device cleans up the PHC-related configurations and resources.
12. The PHC terminates.

Table 28: Scripts Supported for SZTP

Script Type	Interpreter Path	Platform Support
Shell script	#!/bin/sh	All network devices
Python script	#!/usr/bin/python	Network devices running Junos OS with Enhanced Automation Network devices running Junos OS Evolved

SZTP for Network Devices with Dual Routing Engines

Before you upgrade the software on the backup Routing Engine on a network device that run Junos OS software, enable the `secure-ztp provision-backup-re` statement at the `[edit system]` hierarchy on the primary Routing Engine

On network devices that run Junos OS software, enable the `provision-backup-re` statement at the `[edit system]` hierarchy on the primary Routing Engine, so it can bootstrap the backup Routing Engine.

On network devices that run Junos OS Evolved software, enable the `auto-sw-sync` statement at the `[edit system]` hierarchy, so that the primary Routing Engine ensures the same image version is on the backup Routing Engine through either an upgrade or downgrade.

On Junos OS-based systems with dual Routing Engines, the primary Routing Engine downloads the image even if the primary Routing Engine is already running the required image version. The device downloads the image so that the primary Routing Engine is ready to upgrade the backup Routing Engine, if needed.

On Junos OS Evolved-based systems, the primary Routing Engine always keeps a copy of the image it is running.

If you haven't enabled the `synchronize` statement at the `[edit system]` hierarchy or Graceful Restart Engine Switchover (GRES) on the primary Routing Engine, the primary Routing Engine doesn't synchronize the configuration and state to the backup Routing Engine. In this situation, the primary Routing Engine verifies the authenticity of the backup Routing Engine before it synchronizes any data with the backup Routing Engine.

Before the primary Routing Engine provisions the backup Routing Engine, the primary Routing Engine verifies the authenticity of the backup Routing Engine. The primary Routing Engine checks the DevID of the backup Routing Engine to make sure that the backup Routing Engine is a Juniper-authorized Routing Engine.



NOTE: The primary Routing Engine doesn't check whether the backup Routing Engine is authorized to receive information from the primary Routing Engine. Also, the backup Routing Engine doesn't verify authenticity or authorization of the primary Routing Engine.

The primary Routing Engine provisions the backup Routing Engine in the following situations:

- When the primary Routing Engine has bootstrapped using SZTP.
- When the backup Routing Engine is present when the primary Routing Engine is bootstrapping or inserted during the SZTP process.
- When the backup Routing Engine reboots or is replaced.

Once the primary Routing Engine verifies the backup Routing Engine's authenticity and meets the requirements for provisioning, the primary Routing Engine checks the version of software that is running on the backup Routing Engine. If the backup Routing Engine's software version is different from the primary Routing Engine's software version, the primary Routing Engine upgrades the backup Routing Engine to the same software version that the primary Routing Engine is running.

When both Routing Engines are running the same software, the primary Routing Engine synchronizes its configuration with the backup Routing Engine.

RELATED DOCUMENTATION

[Generate Voucher Certificate](#)

[Secure ZTP Quick Start Guide](#)

[Switching between Secure Zero Touch Provisioning and Zero Touch Provisioning | 530](#)

Switching between Secure Zero Touch Provisioning and Zero Touch Provisioning

IN THIS SECTION

- [Overview | 530](#)
- [Benefits | 531](#)
- [Switching between SZTP and ZTP | 531](#)
- [Caveats | 531](#)



NOTE: To see which platforms support Secure Zero Touch Provisioning (SZTP), go to [Feature Explorer](#). In the **Explore Features** section of the Feature Explorer page, select **All Features**. In the **Features Grouped by Feature Family** box, select Secure ZTP. You can also type the name of the feature in the **Search for Features** edit box. See the Release History Table at the end of this topic for more details of how ZTP support has expanded.

Overview

Secure zero-touch provisioning (SZTP) requires additional network infrastructure, such as a secure ZTP server, for provisioning. If you have a secure device with SZTP as its default provisioning method, and don't have the network infrastructure to support SZTP, you can easily switch to ZTP. On the other hand, if your device's default provisioning method is ZTP, and you want to use SZTP for provisioning, you can easily switch to SZTP.

Benefits

- On secure devices, you have the flexibility to switch between using SZTP and ZTP depending on your network infrastructure.

Switching between SZTP and ZTP

See the following table for the Junos OS and Junos OS Evolved commands and the VM Host OS Junos OS commands to use to switch between SZTP and ZTP and vice versa.



NOTE: On MX304 devices without a backup Routing Engine, when you issue the request vmhost zeroize ztp-option secure-(enable | disable) command, you will see the following warning on the console: Backup RE is not present. Zeroize backup RE when it is inserted.

Table 29: Commands for Switching between SZTP and ZTP

Junos OS and Junos OS Evolved	VM Host Junos OS
<pre>request system zeroize ztp-option secure-disable</pre> <p>When you issue this command, the CLI checks to see if the device is a secure device. If the device is secure, the next time the device boots, the device uses ZTP as the provisioning solution. If the device is not secure, the process ends.</p>	<pre>request vmhost zeroize ztp-option secure-disable</pre> <p>When you issue this command, the CLI checks to see if the device is a secure device. If the device is secure, the next time the device boots, the device uses ZTP as the provisioning solution. If the device is not secure, the process ends.</p>
<pre>request system zeroize ztp-option secure-enable</pre> <p>The CLI checks to see if the device is a secure device. If the device is secure, the process ends. The next time the device boots, the device uses SZTP as the provisioning solution. If the device is not a secure device, you will receive an error message that says the device is not secure, and the process ends.</p>	<pre>request vmhost zeroize ztp-option secure-enable</pre> <p>The CLI checks to see if the device is a secure device. If the device is secure, the process ends. The next time the device boots, the device uses SZTP as the provisioning solution. If the device is not a secure device, you will receive an error message that says the device is not secure, and the process ends.</p>

If you don't specify the ztp-option option in either the request system zeroize or request vmhost zeroize command, the secure platform will bootstrap with SZTP as its provisioning solution.

Caveats

- When the device uses ZTP, the SZTP configuration remains on the device, and the SZTP client (phone-home client) runs passively. Once ZTP commits its configuration, the phone-home server configuration is removed.

- If the default ZTP behavior is different from the type of zero-touch provisioning (ZTP or SZTP, for example) you're using, you will need to issue either the `request system zeroize ztp-option secure-(enable | disable)` or

`request vmhost zeroize ztp-option secure-(enable | disable)` command.

- If the current Junos OS or Junos OS Evolved software version on your device supports SZTP, but the software image you're upgrading to doesn't support SZTP, then bootstrapping with SZTP will fail. On devices running Junos OS or VM Host Junos OS, this is not applicable if the device is installed with SZTP as part of its factory default configuration.

RELATED DOCUMENTATION

[Secure ZTP Quick Start Guide](#)

[Secure Zero Touch Provisioning | 519](#)

request system zeroize ztp-option secure

request vmhost zeroize ztp-option secure

8

PART

Phone-Home Client

- [Configure Phone-Home Client | 534](#)
-

Configure Phone-Home Client

IN THIS CHAPTER

- [Obtaining Configurations and Software Image Without User Intervention Using Phone-Home Client | 534](#)
- [Deploying the Phone-home Client and Zero Touch Provisioning on vSRX Virtual Firewall | 538](#)
- [Provision a Virtual Chassis Using the Phone-Home Client | 541](#)

Obtaining Configurations and Software Image Without User Intervention Using Phone-Home Client

IN THIS SECTION

- [Prerequisites | 535](#)
- [Understanding the Phone-Home Client | 535](#)
- [Understanding the Redirect Server Configuration | 535](#)
- [Understanding Interoperability Between the Phone-Home Client and DHCP-Based ZTP | 536](#)
- [Understanding the Phone-Home Client Process | 536](#)
- [Understanding the Configuration File Format for the Phone-Home Client | 537](#)
- [Understanding Pre-Configuration and Post-Configuration Scripts | 537](#)
- [Verifying that the Phone-Home Client Downloaded the Configuration and Software Image | 537](#)

The phone-home client (PHC) enables the device or VM instance to securely obtain bootstrapping data, such as a configuration or software image, with no user intervention other than having to physically connect the device or VM instance to the network.

Prerequisites

PHC depends on the following software and utilities to operate:

- Connectivity to redirect server and phone-home server (PHS)
- DHCP client



NOTE: DHCP-based ZTP is not supported on vSRX Virtual Firewall.

- SLAX support for configuration commits
- Python support
- Curl support
- Factory default configuration
- Mechanism to retrieve device serial number
- SHA1/MD5 utilities to verify software image
- Basic utilities like GREP and AWK

Understanding the Phone-Home Client

PHC enables the device or VM instance to securely obtain bootstrapping data, such as a configuration or software image, with no user intervention other than having to physically connect the device to the network. When the device or VM instance first boots, PHC connects to a redirect server, which then redirects to PHS to get the configuration or software image.

Similar to DHCP-based ZTP, the device or VM instance must be in factory default state in order for PHC to provision the device. If the device is not in factory default state, you can issue the request `system zeroize` command to bring the device back to the factory default state.

Understanding the Redirect Server Configuration

By default, the factory default configuration includes the redirect server URL, which is `https://redirect.juniper.net`.

Understanding Interoperability Between the Phone-Home Client and DHCP-Based ZTP

To avoid conflicts between these two provisioning methods, the following steps are taken when the device or VM instance boots up:



NOTE: DHCP-based ZTP is not supported on vSRX Virtual Firewall.



NOTE: Provisioning does not start if the device is not in factory default mode. If the device is not in factory default mode, issue the `request system zeroize` command.



NOTE: The `request system zeroize` command is not supported on vSRX Virtual Firewall.

1. If the DHCP client receives either partial or complete DHCP options, PHC is terminated, and DHCP-based ZTP attempts to provision the device until it is successful.
2. If the DHCP client does not receive DHCP options, PHC attempts to provision the device until it is successful.

If PHC fails to connect to the redirect server, however, DHCP-based ZTP attempts to provision the device. Both provisioning methods attempt to provision the device until one method is successful.

Understanding the Phone-Home Client Process

The following steps take place when PHC is launched:

1. PHC connects to the redirect server.
2. The device or VM Instance downloads and installs the software image from PHS.
If the software upgrade fails, the process starts over.
3. The device or VM instance reboots, and PHC validates the installed software image when the device comes back online.
4. The device or VM instance downloads the configuration.
5. If a script (either pre-configuration script, post-configuration scripts, or both) was received as part of the configuration, the following happens:



NOTE: PHC supports both Python and shell scripts.

- a. The pre-configuration script is executed.
 - b. The configuration received from the redirect server is committed.
 - c. The post-configuration script is executed.
6. PHC sends a bootstrap-complete message to the PHS.
 7. PHC cleans up the downloaded resources.
 8. The existing phone-home configuration, along with any supporting configuration, is overwritten by the new configuration on the device or VM instance.
 9. If any of the above steps fail, the phone-home process starts over again from the beginning, and a bootstrap failure error message is sent to PHS.

Understanding the Configuration File Format for the Phone-Home Client

PHC supports XML as the file format for the configuration file.

For example, the configuration file format looks like this:

```
<configuration>
[ Configuration in XML format ]
</configuration>
```

Currently, only the `merge` and `override` CLI commands are supported on configurations received by the PHC.

Understanding Pre-Configuration and Post-Configuration Scripts

You can include pre-configuration and post-configuration scripts on PHS in addition to, or instead of, using the Junos OS CLI. Embed the scripts in base64 encoded format. PHC extracts the encoded scripts from the bootstrap information received from PHS, decodes, and then runs the decoded scripts at the appropriate stages of provisioning.

Verifying that the Phone-Home Client Downloaded the Configuration and Software Image

To verify the progress of the phone-home process, you can view the `notification.xml` file on PHS.

Change History Table

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

Release	Description
21.1R1	Starting in Junos OS Release 21.1R1, the phone-home client is supported on vSRX Virtual Firewall.

Deploying the Phone-home Client and Zero Touch Provisioning on vSRX Virtual Firewall

SUMMARY	IN THIS SECTION
	<ul style="list-style-type: none"> Factory Default Configuration on vSRX Virtual Firewall 538 Deploying ZTP on KVM 539 Deploying ZTP on VMWare 539 Deploying ZTP on Amazon Web Services, Google Cloud Platform, and Oracle Cloud Infrastructure 540 Deploying ZTP on Microsoft Azure 541

You can use the phone-home client and ZTP to provide a user-defined configuration file for the vSRX Virtual Firewall. The phone-home client and ZTP are supported on VMWare, KVM (Kernel-based Virtual Machine) hypervisors, and in various deployment environments, such as AWS (Amazon Web Service), GCP (Google Cloud Platform), OCI (Oracle Cloud Infrastructure, and Microsoft Azure.

Factory Default Configuration on vSRX Virtual Firewall

Here's the factory default configuration for the phone-home client:

```

set system services web-management http interface fxp0.0
set system services web-management https system-generated-certificate
set system services web-management https interface fxp0.0
set system name-server 8.8.8.8
set system name-server 8.8.4.4

```

```

set system syslog file messages any any
set system license autoupdate url https://ae1.juniper.net/junos/key_retrieval
set system phone-home server https://redirect.juniper.net
set system phone-home rfc-compliant
set interfaces fxp0 unit 0 family inet dhcp

```



NOTE: You must perform the changes suggested in the 'vSRX Virtual Firewall XML on KVM' and 'vSRX Virtual Firewall virtual machine edit settings in VMware' before the first reboot. This ensures that the correct factory default configuration with PHC commands are loaded during the first boot.

Deploying ZTP on KVM

To deploy ZTP on a KVM, set the **entry name='version' to phone-home-true** in the VM deployment XML file.

For example:

```

<os>
  ...
  <smbios mode='sysinfo' />
</os>
<sysinfo type='smbios'>
  <system>
    <entry name='version'>phonehome=true</entry>
  </system>
</sysinfo>

```

Deploying ZTP on VMWare

To deploy ZTP on VMWare, enable the **Open Virtualization Format (OVF)** setting in the VMWare GUI, and set **phone-home** to **true**.

1. To enable OVF in the VMWare GUI, go to **Edit Virtual Machine Setting | vApp Options | OVF setting : OVF environment transparent | VMWare Tools: enable**.
2. To enable the phone-home client in the VMWare GUI, go to **Edit Virtual Machine Setting | vApp Options | Properties | phone-home true**.

Deploying ZTP on Amazon Web Services, Google Cloud Platform, and Oracle Cloud Infrastructure

To enable ZTP on Amazon Web Services, Google Cloud Platform, and Oracle Cloud Infrastructure, add the following phone-home client configuration in the **CLOUD-INIT USER-DATA** file:

```
system {
  name-server {
    8.8.8.8;
    8.8.4.4;
  }
  syslog {
    file messages {
      any any;
    }
  }
  services {
    ssh;
    web-management {
      http {
        interface fxp0.0;
      }
      https {
        system-generated-certificate;
        interface fxp0.0;
      }
    }
  }
}
license {
  autoupdate {
    url https://ae1.juniper.net/junos/key_retrieval;
  }
}
phone-home {
  server https://redirect.juniper.net;
  rfc-compliant;
}

interfaces {
  fxp0 {
```

```

        unit 0 {
            family inet {
                dhcp;
            }
        }
    }
}

```

Deploying ZTP on Microsoft Azure

To enable ZTP on Microsoft Azure, add the following phone-home client configuration in the **write_files** section of the **CLOUD_INIT_CONFIG.JSON** file:

```

{
  "content": "configure\nset system name-server 8.8.8.8\nset system name-server 8.8.4.4\nset\nsystem phone-home server\nhttps://redirect.juniper.net\nset system phone-home rfc-compliant\nset interfaces fxp0 unit 0\nfamily inet dhcp\ncommit\n",
  "path": "/var/tmp/test_config"
},

```

Provision a Virtual Chassis Using the Phone-Home Client

SUMMARY

Phone-home provisioning on a Virtual Chassis is a form of zero-touch provisioning (ZTP). The phone-home client (PHC) on the Virtual Chassis gets bootstrap information over the network from a phone-home server (PHS) and provisions the Virtual Chassis. The only user intervention required on the client side is to physically wire the Virtual Chassis members together and connect any port on the Virtual Chassis to the network.

IN THIS SECTION

- [Overview of Phone-Home Provisioning for a Virtual Chassis | 542](#)
- [How To Enable Phone-Home Provisioning on a Virtual Chassis | 544](#)
- [Phone-Home Process on a Virtual Chassis | 545](#)
- [Phone-Home Provisioning Status Notifications | 550](#)

- [Verify Virtual Chassis Status After Phone-Home Provisioning | 552](#)
- [Troubleshoot Phone-Home Provisioning Issues | 553](#)

Overview of Phone-Home Provisioning for a Virtual Chassis

IN THIS SECTION

- [Benefits of Phone-Home Provisioning on a Virtual Chassis | 542](#)
- [Overview of the Phone-Home Provisioning Process on a Virtual Chassis | 543](#)

With phone-home provisioning, a phone-home client (PHC) on a device initially provisions the device with a software image and configuration from a central network management data source called the phone-home server (PHS), requiring little or no user intervention at the remote site.

A Virtual Chassis consists of a set of devices interconnected together using ports called Virtual Chassis ports (VCPs). You configure and manage the Virtual Chassis as a single device. Starting with Junos OS Release 20.3R1, we've made extensions to the phone-home provisioning process for a standalone device so it can also work on a Virtual Chassis. The PHC on a Virtual Chassis requires extra steps to coordinate and manage bootstrapping the member devices.

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

You can check [Feature Explorer](#) and search for **phone-home** to see the Virtual Chassis platforms that support phone-home provisioning.

Benefits of Phone-Home Provisioning on a Virtual Chassis

- Simplifies provisioning by launching the process automatically from the remote site, while securely obtaining bootstrap information from a central management system (the PHS) on your network or in the cloud.
- Doesn't require in-depth experience with the Junos OS CLI to coordinate the provisioning of multiple devices that make up a Virtual Chassis.

Overview of the Phone-Home Provisioning Process on a Virtual Chassis

On a Virtual Chassis that supports phone-home provisioning, for the process to work, you must set up the Virtual Chassis according to the requirements outlined in ["How to Enable Phone-Home Provisioning on a Virtual Chassis" on page 544](#).

When the Virtual Chassis initially forms, the PHC process starts up automatically on the Virtual Chassis primary member and takes it from there:

1. The PHC connects to a PHS.

The PHC sends a provisioning request to a default redirect server URL, <https://redirect.juniper.net>, which redirects the request to an available PHS controlled by your network administrator or NMS. This step is the same as phone-home provisioning on a single device.

2. The PHS responds to the PHC provisioning request with the bootstrapping information, which includes the intended Virtual Chassis topology, software image, and configuration.

3. The PHC provisions the Virtual Chassis as specified by the PHS.

Provisioning includes steps such as:

- Validate the Virtual Chassis topology.
- Upgrade the software image sequentially on all of the member devices if needed.
- Run any pre-configuration or post-configuration staging scripts.
- Commit a new configuration on the Virtual Chassis.

The PHC sends status notifications to the PHS during the bootstrapping process, so the network administrator can verify the process completes successfully.

The PHC also logs status locally in the system log files on the Virtual Chassis. If needed, you can view log files in the Junos OS CLI, and use Junos OS CLI commands to see Virtual Chassis and VCP connection status.

SEE ALSO

[Obtaining Configurations and Software Image Without User Intervention Using Phone-Home Client](#)
| 534

[Zero Touch Provisioning](#) | 463

How To Enable Phone-Home Provisioning on a Virtual Chassis

On a Virtual Chassis that supports phone-home provisioning, if you set up the Virtual Chassis according to the steps listed here, a phone-home client (PHC) process starts up automatically on the Virtual Chassis primary member.

To enable phone-home provisioning on a Virtual Chassis:

1. Ensure that all Virtual Chassis members have the factory-default configuration and are powered off.

You can run the `request system zeroize` Junos OS CLI command to return a device to its factory-default state.



NOTE: The Virtual Chassis can't be a mixed-mode Virtual Chassis because mixed mode is never set in the factory-default configuration.

2. Interconnect the Virtual Chassis members in a ring topology using only dedicated or default-configured Virtual Chassis ports (VCPs) on each member device.

Keep in mind that the PHC process works only if the Virtual Chassis is initially formed with VCPs that do not need to be explicitly configured (dedicated VCPs or ports that are VCPs in the factory-default configuration). See *VCP Options by Switch Type* for details on which ports are dedicated and default-configured VCPs on different devices that support Virtual Chassis. See the hardware guide for the device to locate those ports on the device.

3. Connect the Virtual Chassis management interface (`me0`) or any network-facing port on any Virtual Chassis member to the network.

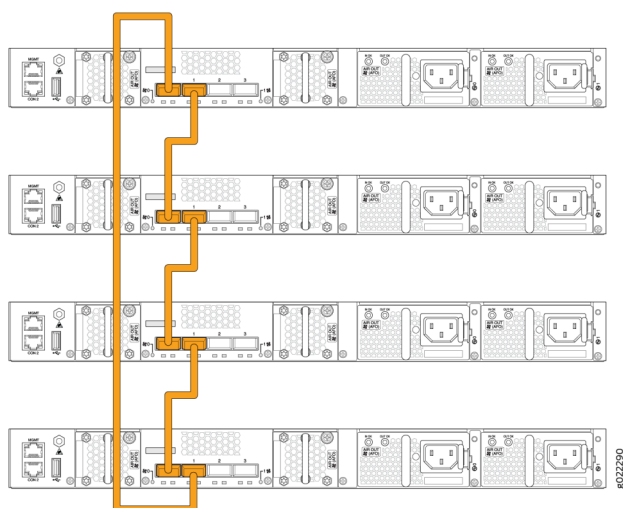
After the PHC starts up on the Virtual Chassis, it uses this connection to access a PHS over the network and retrieve the bootstrapping information for this Virtual Chassis.

For details about how the management interfaces work on a Virtual Chassis, see *Understanding Global Management of a Virtual Chassis*.

4. Power on the members of the Virtual Chassis.

Figure 25 on page 545 shows an example of a Virtual Chassis topology that can support phone-home provisioning—a four-member EX4300 Virtual Chassis cabled in a ring topology using default-configured VCPs (in this case, two of the 40-Gigabit Ethernet QSFP+ ports on each device).

Figure 25: Sample Virtual Chassis That Can Support Phone-Home Provisioning



Usually you don't need to do anything else for the phone-home provisioning process to proceed and complete successfully. If you don't see successful completion status or the Virtual Chassis isn't up and operating as expected at the end of the process, read on to learn details about how the PHC works to help troubleshoot the issues.

SEE ALSO

Understanding Virtual Chassis Components

Virtual Chassis Overview for Switches

Phone-Home Process on a Virtual Chassis

IN THIS SECTION

- [Startup and Request Provisioning Information from PHS | 546](#)
- [Bootstrap Virtual Chassis Members | 548](#)
- [Apply Scripts and New Configuration on the Virtual Chassis | 549](#)
- [Provisioning Process Completion | 550](#)

Phone-home provisioning on a Virtual Chassis is an extension of the standalone device phone-home support described in ["Obtaining Configurations and Software Image Without User Intervention Using Phone-Home Client" on page 534](#). The PHC performs additional steps to manage bootstrapping the member devices that make up the Virtual Chassis.

The PHC process on a Virtual Chassis also requires the same software tools and utilities that standalone devices require for PHC to work. For example, the phone-home process needs DHCP client support to facilitate the network connection to the PHS in the same way as for a single device, and verifies a downloaded software image using the same checksum utilities. See ["Prerequisites" on page 535](#) for a list of these general PHC requirements.

Phone-home provisioning starts up automatically on a Virtual Chassis on the client side after you've performed the tasks in ["How To Enable Phone-Home Provisioning on a Virtual Chassis" on page 544](#) and if the Virtual Chassis meets the conditions described in *Requirements for Phone-Home Provisioning to Work for a Virtual Chassis*.

The provisioning process steps are grouped into the stages described in this section.

Startup and Request Provisioning Information from PHS

In the startup and request stage:

1. The Virtual Chassis boots up in factory-default or zeroized state as a nonprovisioned Virtual Chassis, and elects the initial primary and backup members. (See *Understanding How the Primary in a Virtual Chassis Is Elected*.)
2. The PHC starts up on the Virtual Chassis primary member, connects to the default redirect server (<https://redirect.juniper.net>) and sends a bootstrap request for the device. The redirect server redirects the PHC to an available PHS.
3. The PHC receives the response from PHS, starts to discover what Virtual Chassis members are connected, and prepares to provision the Virtual Chassis. The PHS response includes:
 - Virtual Chassis topology information—At a minimum, this part of the response indicates the device is expected to be part of a Virtual Chassis; otherwise, the PHC provisions only the primary member as a standalone device.

The response might additionally have full topology information, which includes the serial IDs of all the members the network administrator expects to be in the Virtual Chassis.

- Software image upgrade information—Includes a path to the intended software image and image verification details.
- Pre-configuration and post-configuration script information—Includes any staging scripts the network administrator needs the PHC to run before or after applying the new configuration.
- Configuration information—Includes the intended Virtual Chassis configuration and the method to apply that configuration.

The PHC must receive minimum required topology information to recognize it should provision a Virtual Chassis. Otherwise, the PHC defaults to provisioning only the primary member as a standalone device.

The PHC extensions for Virtual Chassis support two provisioning modes—the default mode and a more strict mode the PHS can specify in the response:

- By default, the PHC provisions any members it detects in the VC at the time it receives the PHS response. If the PHC encounters an error when bootstrapping a particular member, it moves on to bootstrap the next member or continues to the next provisioning step.
- If the PHS specifies the strict mode option in the response, the response must also include the full Virtual Chassis topology information. Provisioning succeeds only if the PHC finds and successfully bootstraps all of the same members listed in the PHS response. If the PHC doesn't detect all of the intended members or provisioning fails for any of them, the PHC restarts the process from the beginning to resend the provisioning request to another available PHS.



NOTE: The PHS can include the Virtual Chassis member serial IDs in the response in either mode. However, in the default mode, bootstrapping the Virtual Chassis can succeed even if the PHC doesn't detect all the members in the PHS response or if the PHS response doesn't include member details at all.

In this step, if the response includes full Virtual Chassis topology information and indicates to use strict provisioning mode, the PHC validates what it finds in the Virtual Chassis locally against the Virtual Chassis member information from the PHS response.

[Table 30 on page 547](#) summarizes the actions the PHC takes starting in this step based on the type of topology information it receives in the PHS response, the provisioning mode, and the Virtual Chassis members that the PHS discovers locally.

Table 30: PHC Actions Based on Topology Information in PHS Response

Virtual Chassis Topology Information from PHS	Default Mode Actions	Strict Mode Actions If PHS Requests This Option
No topology information provided	Try to provision as standalone device.	N/A (This mode can be specified only with full topology information)

Table 30: PHC Actions Based on Topology Information in PHS Response (Continued)

Virtual Chassis Topology Information from PHS	Default Mode Actions	Strict Mode Actions If PHS Requests This Option
Minimum required topology information for a Virtual Chassis	Discover members and proceed to provision Virtual Chassis with found members.	N/A (This mode can be specified only with full topology information)
Full topology information for a Virtual Chassis, including serial IDs for all intended Virtual Chassis members	Discover members. If member list doesn't match PHS response, proceed anyway to provision Virtual Chassis with found members.	Bootstrap intended Virtual Chassis members from PHS response. Detect members; if all expected members are present and up, provisioning succeeds. Otherwise retry to bootstrap and detect members that failed the process. After member detection timeout with failure to detect all expected members, report error and restart process contacting another PHS to re-request provisioning.

4. If the PHC proceeds to provision the devices in the Virtual Chassis, at this point the PHC commits some temporary changes to the Virtual Chassis configuration to enable smooth bootstrapping of all VC members.

For example, the PHC makes sure the Virtual Chassis primary and backup member roles don't change while the PHC is upgrading the software image on all the members.

Bootstrap Virtual Chassis Members

In this stage, the PHC bootstraps the Virtual Chassis, which includes installing the software image on and rebooting all of the members.

1. The PHC on the primary member compares the bootstrap information in the PHS response with what's on the Virtual Chassis to see if it needs to upgrade the software image. If the versions match, the PHC skips the remaining steps in this stage.
2. If the PHC needs to upgrade the software image, the PHC uses the bootstrap information in the PHS response (image filename and checksum information) to download and validate the image.

If the download operation fails, the PHC retries until it succeeds. (This behavior is the same for standalone device or Virtual Chassis phone-home provisioning.)

3. The PHC proceeds to install and reboot the Virtual Chassis members based on the member roles as follows in this order:
 - a. Linecard members—Installs the image on the linecard role members sequentially (in member ID order), and then reboots them all at the same time.
 - b. Backup member—Installs the image on the backup member and reboots it.
 - c. Primary member—Installs the image on the primary member, synchronizes the current PHC Virtual Chassis bootstrap state to the backup member, and triggers the primary member to reboot.

As the upgraded members boot up, the PHC checks that they are up and running again. This action is called *member detection* in log messages and status notifications. If the PHC fails to detect a member within a default member detection timeout, the PHC notifies the PHS of the error. See ["Startup and Request Provisioning Information from PHS" on page 546](#) for the actions the PHC takes by default or if the PHS specified strict provisioning.

4. While the old primary member is rebooting, the original primary isn't available, so the Virtual Chassis switches primary role to the backup member. The Virtual Chassis also elects a new backup member at this time.
5. The PHC starts up on the new primary member (the original backup member) and resumes the Virtual Chassis bootstrap procedure from the PHC state inherited from the old primary.
When the old primary finishes booting and rejoins the Virtual Chassis, it is initially in linecard role but then assumes the backup role to the new primary member.
6. When the PHC detects this last member is up and running, the provisioning process continues to the next stage to apply pre-configuration or post-configuration scripts and the new configuration to the Virtual Chassis.



NOTE: Avoid having pre-configuration scripts, post-configuration scripts, or the new configuration make any changes that might cause the Virtual Chassis to assign new member roles or elect new primary and backup members during the provisioning process. Otherwise, provisioning might fail with unpredictable results.

Apply Scripts and New Configuration on the Virtual Chassis

The PHS response might include pre-configuration and post-configuration scripts the network administrator needs the PHC to run on the virtual Chassis before or after applying the new configuration. Phone-home provisioning supports Python or shell scripts and only XML format for the configuration.

The PHS response also provides the Junos OS configuration for PHC to commit on the member devices in the Virtual Chassis.

A Virtual Chassis operates as if it's a single device, so the PHC performs these steps on the Virtual Chassis as a whole:

1. Runs any specified pre-configuration scripts from PHS.
2. Applies and commits the new configuration from PHS.
3. Runs any specified post-configuration scripts from PHS.

Provisioning Process Completion

To complete the phone-home provisioning process, the PHC logs that the process completed successfully and sends a bootstrap completion notification to the PHS.

The PHC doesn't run again unless you return the device or Virtual Chassis back to the factory-default state and have all other required conditions to trigger phone-home provisioning.

See *Requirements for Phone-Home Provisioning to Work for a Virtual Chassis* for details.

Phone-Home Provisioning Status Notifications

The PHC logs status information locally in the system log (`/var/log/messages`) on the Virtual Chassis and sends status notifications to the PHS to report the progress of the provisioning process. These messages signal when the PHC completes the different provisioning stages and help you troubleshoot issues if the process doesn't complete successfully. See ["Phone-Home Process on a Virtual Chassis" on page 545](#) for the steps the PHC performs in each stage of Virtual Chassis provisioning.

Some PHC status messages are general and apply either for single device or Virtual Chassis provisioning.

Notification messages that are specific to a particular Virtual Chassis member include:

- The member ID
- The member's serial ID
- The member's current role in the Virtual Chassis—Master, Backup, or Linecard

Virtual Chassis member-specific notifications have the following format:

```
vc-member [memberID:serialID:role] message
```

For example:

```
vc-member [2:AA1234567890:Backup] Successfully installed downloaded image. Initiating member
reboot.
```

Phone-home process notifications consist of a notification type and message. [Table 31 on page 551](#) lists notifications that are specific to the phone-home provisioning stages on a Virtual Chassis. Notification types with the `vc-member` keyword include Virtual Chassis member-specific information.

Table 31: PHC Notifications for Virtual Chassis Provisioning Steps

Notification Type	Notification Message
<code>vc-member-image-installed</code>	<ul style="list-style-type: none"> • Successfully installed downloaded image. Initiating image installation on next member. • Successfully installed downloaded image. Initiating member reboot.
<code>vc-member-image-installation-failed</code>	<ul style="list-style-type: none"> • Image failed to install on the member. Giving up and trying a different phone-home-server.
<code>vc-member-reboot-initiated</code>	<ul style="list-style-type: none"> • Reboot initiated for Line Card members. Waiting for the members to come back up.
<code>vc-member-upgrade-success</code>	<ul style="list-style-type: none"> • No upgrade required. • Successfully upgraded. • Member detected and successfully upgraded.
<code>vc-member-upgrade-failed</code>	<ul style="list-style-type: none"> • Upgrade failed !!! • Member detected but upgrade failed !!!
<code>vc-member-detection-failed</code>	<ul style="list-style-type: none"> • Did not come up post image upgrade <p>NOTE: This message means the PHC installed the new image and initiated a reboot of the Virtual Chassis member, but the PHC didn't detect that the member came up again within a prescribed member detection timeout.</p>

Table 31: PHC Notifications for Virtual Chassis Provisioning Steps *(Continued)*

Notification Type	Notification Message
vc-bootstrap-failed	<ul style="list-style-type: none"> VC member bootstrap failure detected with Strict provisioning set. <p>NOTE: This message mean the PHC upgraded the expected linecard role members successfully, but after rebooting them, PHC didn't detect that all members came up again within a prescribed member detection timeout.</p> <ul style="list-style-type: none"> VC with detection failed members and Strict provisioning set. <p>NOTE: This message means the PHC failed to detect one or more members after upgrading and rebooting all of them, and upon checking again, finds that one or more of them still failed to come up.</p> <p>With strict provisioning mode, PHC must successfully bootstrap all intended members for the provisioning process to signal successful completion.</p>

Verify Virtual Chassis Status After Phone-Home Provisioning

IN THIS SECTION

- Purpose | 552
- Action | 552

Purpose

Check the running status of the Virtual Chassis after PHC provisioning.

Action

Enter the `show virtual-chassis` command using the Junos OS CLI.

For example:

```
{master:1}
user@device>
Virtual Chassis ID: xxxx.xxxx.xxxx
Virtual Chassis Mode: Enabled
```

Member ID	Status	Serial No	Model	Mstr prio	Role	Mixed Mode	Route Mode	Neighbor ID	List Interface
0 (FPC 0)	Prsnt	...	ex3400-24p	128	Backup	N	VC	2	vcp-255/1/0
								1	vcp-255/1/1
1 (FPC 1)	Prsnt	...	ex3400-24p	128	Master*	N	VC	0	vcp-255/1/0
								2	vcp-255/1/1
2 (FPC 2)	Prsnt	...	ex3400-48p	128	Linecard	N	VC	0	vcp-255/1/0
								1	vcp-255/1/1

Member ID for next new member: 3 (FPC 3)

Troubleshoot Phone-Home Provisioning Issues

To troubleshoot PHC problems during the provisioning process:

- Use utilities on the PHS side specific to your network management system to check device, Virtual Chassis, and connection status, or display phone-home process notifications (see ["Phone-Home Provisioning Status Notifications" on page 550](#)).
- Make sure the Virtual Chassis management or network interface is connected to the network and can connect to a PHS.
- If the PHS specified the strict mode option, verify the Virtual Chassis member serial IDs on the phone-home server side match the member devices you're interconnecting on the client side at the remote site.
- Look for error and status messages in the syslog file on the Virtual Chassis.

For example, syslog status messages can show that the ZTP client is trying to provision the device instead of or in addition to the PHC. Upon startup with the factory-default configuration on either a standalone device or a Virtual Chassis primary member, both the PHC and the DHCP-based ZTP process (see ["Zero Touch Provisioning" on page 463](#)) start running automatically. ZTP proceeds if DHCP ZTP options are configured, which can cause unexpected provisioning behavior because ZTP isn't supported for a Virtual Chassis. To trigger only phone-home provisioning, your DHCP system administrator can make sure the ZTP-specific options are not set on the DHCP server for devices intended to be in a Virtual Chassis under PHS management.

- Check the configuration on the Virtual Chassis after provisioning using the `show configuration` CLI command.

RELATED DOCUMENTATION

[Obtaining Configurations and Software Image Without User Intervention Using Phone-Home Client](#)
| [534](#)

Understanding Virtual Chassis Components

[Zero Touch Provisioning](#) | [463](#)

9

PART

Automatic Installation of Configuration Files

- [Configure Automatic Installation of Configuration Files | 556](#)
-

Configure Automatic Installation of Configuration Files

IN THIS CHAPTER

- [Understanding Autoinstallation of Configuration Files \(Junos OS\) | 556](#)
- [Configuring Autoinstallation of Configuration Files \(Junos OS\) | 567](#)
- [Configuring Autoinstallation of Configuration Files on ACX Series \(Junos OS\) | 582](#)

Understanding Autoinstallation of Configuration Files (Junos OS)

IN THIS SECTION

- [Autoinstallation Overview | 556](#)
- [Autoinstallation Process on Satellite Devices in a Junos Node Unifier Group | 564](#)

Autoinstallation is an automated process and does not require any specific configuration on a device. To simplify the process, you can specify one or more interfaces, protocols, and configuration servers to be used for autoinstallation.

Autoinstallation Overview

IN THIS SECTION

- [Automatic Installation of Configuration Files | 557](#)
- [Supported Autoinstallation Interfaces and Protocols | 558](#)
- [Typical Autoinstallation Process on a New Device | 558](#)

- Typical Uses for Autoinstallation | 562
- Autoinstallation Configuration Files and IP Addresses | 562
- Typical Autoinstallation Process on a New Switch | 563

If you are setting up many devices, autoinstallation can help automate the configuration process by loading configuration files onto new or existing devices automatically over the network. You can use either the J-Web configuration editor or the CLI configuration editor to configure a device for autoinstallation.

Autoinstallation provides automatic configuration for a new device that you connect to the network and turn on, or for a device configured for autoinstallation. The autoinstallation process begins any time a device is powered on and cannot locate a valid configuration file in the CompactFlash (CF) card. Typically, a configuration file is unavailable when a device is powered on for the first time, or if the configuration file is deleted from the CF card. The autoinstallation feature enables you to deploy multiple devices from a central location in the network.

For the autoinstallation process to work, you must store one or more host-specific or default configuration files on a configuration server in the network and have a service available—typically Dynamic Host Configuration Protocol (DHCP)—to assign an IP address to the device.

Autoinstallation takes place automatically when you connect an Ethernet or serial port on a new Juniper Networks device to the network and power on the device. To simplify the process, you can explicitly enable autoinstallation on a device and specify a configuration server, an autoinstallation interface, and a protocol for IP address acquisition.

This section contains the following topics:

Automatic Installation of Configuration Files

On SRX Series Firewalls, you can specify a remote server where configuration files are located. If a configuration file cannot be found on the device's CompactFlash card, the device automatically retrieves the configuration file from this remote server. For security purposes, you can encrypt these remote files using the DES cipher, and once they have been retrieved, the device decrypts them for use on the server.

To encrypt the files, we recommend the OpenSSL tool. You can get the OpenSSL tool at <http://www.openssl.org/>. To encrypt the file, use the following syntax:

```
% openssl enc -des -k passphrase -in original-file -out encrypted-file
```

- *passphrase*—Passphrase used to encrypt the configuration file. The passphrase should be the name of the file without the path information or file extension.
- *original-file*—Unencrypted configuration file.
- *encrypted-file*—Name of the encrypted configuration file.

For example, if you are encrypting the active configuration file `juniper.conf.gz`, the passphrase is `juniper.conf`. The openssl syntax used to encrypt the file is:

```
% openssl enc -des -k juniper.conf -in juniper.conf.gz -out juniper.conf.gz.enc
```

Supported Autoinstallation Interfaces and Protocols

Before autoinstallation on a device can take place, the device must acquire an IP address. The protocol or protocols you choose for IP address acquisition determine the device interface to connect to the network for autoinstallation. The device detects the connected interface and requests an IP address with a protocol appropriate for the interface. Autoinstallation is supported over an Ethernet LAN interface or a serial LAN or WAN interface. [Table 32 on page 558](#) lists the protocols that the device can use on these interfaces for IP address acquisition.

Table 32: Interfaces and Protocols for IP Address Acquisition During Autoinstallation

Interface and Encapsulation Type	Protocol for Autoinstallation
Ethernet LAN interface with High-Level Data Link Control (HDLC)	DHCP, BOOTP, or Reverse Address Resolution Protocol (RARP)
Serial WAN interface with HDLC	Serial Line Address Resolution Protocol (SLARP)
Serial WAN interface with Frame Relay	BOOTP

If the server with the autoinstallation configuration file is not on the same LAN segment as the new device, or if a specific device is required by the network, you must configure an intermediate device directly attached to the new device through which the new device can send Trivial File Transfer Protocol (TFTP), BOOTP, and Domain Name System (DNS) requests. In this case, you specify the IP address of the intermediate device as the location to receive TFTP requests for autoinstallation.

Typical Autoinstallation Process on a New Device

When a device is powered on for the first time, it performs the following autoinstallation tasks:

1. The new device sends out DHCP, BOOTP, RARP, or SLARP requests on each connected interface simultaneously to obtain an IP address.

If a DHCP server responds, it provides the device with some or all of the following information:

- An IP address and subnet mask for the autoinstallation interface.
- The location of the TFTP (typically), Hypertext Transfer Protocol (HTTP), or FTP server on which the configuration file is stored.
- The name of the configuration file to be requested from the TFTP server.
- The IP address or hostname of the TFTP server.

If the DHCP server provides only the hostname, a DNS server must be available on the network to resolve the name to an IP address.

- The IP address of an intermediate device if the configuration server is on a different LAN segment from the new device.
2. After the new device acquires an IP address, the autoinstallation process on the device attempts to download a configuration file in the following ways:
 - a. If the DHCP server specifies the host-specific configuration file (boot file) *hostname.conf*, the device uses that filename in the TFTP server request. (In the filename, *hostname* is the hostname of the new device.) The autoinstallation process on the new device makes three unicast TFTP requests for *hostname.conf*. If these attempts fail, the device broadcasts three requests to any available TFTP server for the file.
 - b. If the new device cannot locate *hostname.conf*, the autoinstallation process unicasts or broadcasts TFTP requests for a default device configuration file called *network.conf*, which contains hostname-to-IP address mapping information, to attempt to find its hostname.
 - c. If *network.conf* contains no hostname entry for the new device, the autoinstallation process sends out a DNS request and attempts to resolve the new device's IP address to a hostname.
 - d. If the new device can determine its hostname, it sends a TFTP request for the *hostname.conf* file.
 - e. If the new device is unable to map its IP address to a hostname, it sends TFTP requests for the default configuration file *router.conf*.
 3. After the new device locates a configuration file on a TFTP server, autoinstallation downloads the file, installs the file on the device, and commits the configuration.



NOTE:

- If you configure the DHCP server to provide only the TFTP server hostname, add an IP address-to-hostname mapping entry for the TFTP server to the DNS database file on the DNS server in the network.
- If the new device is not on the same network segment as the DHCP server (or other device providing IP address resolution), configure an existing device as an intermediate to receive TFTP and DNS requests and forward them to the TFTP server and the DNS server. You must configure the LAN or serial interface on the intermediate device with the IP addresses of the hosts providing TFTP and DNS service. Connect this interface to the new device.



NOTE: Starting in Junos OS Release 15.1X49-D60 and in Junos OS Release 17.3R1, on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Firewalls, some of the factory-default configurations are changed.

- The `name-server` statement, used to configure one or more Domain Name System (DNS) name servers, is changed to 8.8.8.8 and 8.8.4.4. Previously, it was 208.67.222.222 and 208.67.220.220.
- A new system service, NETCONF service over SSH, is introduced at the `[edit system services]` hierarchy:

```
edit system services netconf ssh
```

- The following configuration setting for HTTPS (secure management) access using the J-Web interface is changed. Now, there is no need to specify the interface details for J-Web management. With this configuration, you can manage the device from any interface through HTTPS.

```
edit system services web-management https interface [irb.0]
```

- A license autoupdate URL (https://ae1.juniper.net/junos/key_retrieval) is now supported under the `[edit system]` hierarchy:

```
license {
  autoupdate {
    url https://ae1.juniper.net/junos/key_retrieval;
```



```
    }
}
```

- A new system log configuration is introduced to configure system log messages to record all commands entered by users and all authentication or authorization attempts under the [edit system] hierarchy:

```
syslog {
    archive size 100k files 3;
    user * {
        any emergency;
    }
    file messages {
        any notice;
        authorization info;
    }
    file interactive-commands {
        interactive-commands any;
    }
}
```

The above factory-default configurations are also applicable on SRX380 Series Firewalls.



NOTE: On SRX300, SRX320, SRX340, SRX345, SRX380, and SRX550M Firewalls, telnet and xnm-clear-text are not part of system services in factory-default configurations.



NOTE: In Junos OS Release 15.1X49-D40 and earlier, configuring autoinstallation using USB and Layer Ethernet switching was supported on the same interface. However, the command caused improper installation of the interface-related configurations. Layer 2 Ethernet switching is not supported on the same interface for SRX300, SRX320, SRX340, SRX345, SRX380, and SRX550M Firewalls.

The system autoinstallation interfaces <interface names> command and the set interface <interface names> unit 0 family ethernet-switching command cannot be configured on the same interface.



NOTE: USB auto-installation is not supported on SRX1500 Firewalls and vSRX Virtual Firewall instances.

Autoinstallation is the automatic configuration of a device over the network from a preexisting configuration file that you create and store on a configuration server—typically a Trivial File Transfer Protocol (TFTP) server. You can use autoinstallation to configure new devices automatically and to deploy multiple devices from a central location in the network.

You enable autoinstallation so that the switches in your network implement autoinstallation when they are powered on. To configure autoinstallation, you specify a configuration server, an autoinstallation interface, and a protocol for IP address acquisition.



NOTE: The QFX5200 switches only work with HTTP for autoinstallation. TFTP and FTP protocols are not supported.

Typical Uses for Autoinstallation

Typical uses for autoinstallation of the software include:

- To deploy and update multiple devices from a central location in the network.
- To update a device—Autoinstallation occurs when a device that has been manually configured for autoinstallation is powered on.

Autoinstallation Configuration Files and IP Addresses

For the autoinstallation process to work, you must store one or more host-specific or default configuration files on a configuration server in the network and have a service available—typically Dynamic Host Configuration Protocol (DHCP)—to assign an IP address to the switch.

You can set up the following configuration files for autoinstallation on the switch:

- **network.conf**—Default configuration file for autoinstallation, in which you specify IP addresses and associated hostnames for devices on the network.
- **switch.conf**—Default configuration file for autoinstallation with a minimum configuration sufficient for you to telnet to the device and configure it manually.
- **hostname.conf**—Host-specific configuration file for autoinstallation on a device that contains all the configuration information necessary for the switch. In the filename, **hostname** is replaced with the hostname assigned to the switch.

If the server with the autoinstallation configuration file is not on the same LAN segment as the new device, or if a specific device is required by the network, you must configure an intermediate device directly attached to the new switch, through which the new switch can send TFTP, Boot Protocol (BOOTP), and Domain Name System (DNS) requests. In this case, you specify the IP address of the intermediate device as the location to receive TFTP requests for autoinstallation.

Typical Autoinstallation Process on a New Switch

When the switch configured for autoinstallation is powered on, it performs the following autoinstallation tasks:

1. The switch sends out DHCP or BOOTP requests on each connected interface simultaneously to obtain an IP address.

If a DHCP server responds to these requests, it provides the switch with some or all of the following information:

- An IP address and subnet mask for the autoinstallation interface.
- The location of the (typically) TFTP server, Hypertext Transfer Protocol (HTTP) server, or FTP server on which the configuration file is stored.
- The name of the configuration file to be requested from the TFTP server.
- The IP address or hostname of the TFTP server.

If the DHCP server provides the server's hostname, a DNS server must be available on the network to resolve the name to an IP address.

- The IP address of an intermediate device if the configuration server is on a different LAN segment from the switch.
2. After the switch acquires an IP address, the autoinstallation process on the switch attempts to download a configuration file in the following ways:
 - a. If the DHCP server specifies the host-specific configuration file **hostname.conf**, the switch uses that filename in the TFTP server request. The autoinstallation process on the new switch makes three unicast TFTP requests for **hostname.conf**. If these attempts fail, the switch broadcasts three requests to any available TFTP server for the file.
 - b. If the switch does not locate a **hostname.conf** file, the autoinstallation process sends three unicast TFTP requests for a **network.conf** file that contains the switch's hostname-to-IP-address mapping information. If these attempts fail, the switch broadcasts three requests to any available TFTP server for the file.

- c. If the switch fails to find a **network.conf** file that contains a hostname entry for the switch, the autoinstallation process sends out a DNS request and attempts to resolve the switch's IP address to a hostname.
 - d. If the switch determines its hostname, it sends a TFTP request for the **hostname.conf** file.
 - e. If the switch is unable to map its IP address to a hostname, it sends TFTP requests for the default configuration file **switch.conf**. The TFTP request procedure is the same as for the **network.conf** file.
3. After the switch locates a configuration file on a TFTP server, the autoinstallation process downloads the file, installs the file on the switch, and commits the configuration.



NOTE: Please refer to the product [Data Sheets](#) for details, or contact your Juniper Account Team or Juniper Partner. Please refer to the [Juniper Licensing Guide](#) for general information about License Management.

Autoinstallation Process on Satellite Devices in a Junos Node Unifier Group

IN THIS SECTION

- [Supported Autoinstallation Interfaces and Protocols | 565](#)
- [Typical Autoinstallation Process on a New Router | 565](#)

Autoinstallation provides automatic configuration for a new router that you connect to the network and power on, or for a router configured for autoinstallation. The autoinstallation process begins any time a router is powered on and cannot locate a valid configuration file in the CompactFlash card. Typically, a configuration file is unavailable when a router is powered on for the first time, or if the configuration file is deleted from the CompactFlash card. The autoinstallation feature enables you to deploy multiple routers from a central location in the network.

For the autoinstallation process to work, you must store one or more host-specific or default configuration files on a configuration server in the network and have a service available—typically Dynamic Host Configuration Protocol (DHCP)—to assign an IP address to the router.

Autoinstallation takes place automatically when you connect an Ethernet interface on a new Juniper Networks router to the network and power on the router. To simplify the process, you can explicitly enable autoinstallation on a router and specify a configuration server, an autoinstallation interface, and a protocol for IP address acquisition.

This topic describes:

Supported Autoinstallation Interfaces and Protocols

Before autoinstallation on a router can take place, the router must acquire an IP address or a USB key. The protocol or protocols you choose for IP address acquisition determine the router interface to connect to the network for autoinstallation. The router detects the connected interface and requests an IP address with a protocol appropriate for the interface. Autoinstallation is supported over an Ethernet LAN interface. For IP address acquisition, the JNU satellite router uses DHCP, BOOTP, or Reverse Address Resolution Protocol (RARP) on an Ethernet LAN interface.

If the server with the autoinstallation configuration file is not on the same LAN segment as the new router, or if a specific router is required by the network, you must configure an intermediate router directly attached to the new router, through which the new router can send HTTP, FTP, Trivial File Transfer Protocol (TFTP), BOOTP, and Domain Name System (DNS) requests. In this case, you specify the IP address of the intermediate router as the location to receive HTTP, FTP, or TFTP requests for autoinstallation.

Typical Autoinstallation Process on a New Router

When a router is powered on for the first time, it performs the following autoinstallation tasks:

1. The new router sends out DHCP, BOOTP, or RARP requests on each connected interface simultaneously to obtain an IP address.

If a DHCP server responds, it provides the router with some or all of the following information:

- An IP address and subnet mask for the autoinstallation interface.
- The location of the TFTP (typically), HTTP, or FTP server on which the configuration file is stored.
- The name of the configuration file to be requested from the HTTP, FTP, or TFTP server.
- The IP address or hostname of the HTTP, FTP, or TFTP server.

If the DHCP server provides only the hostname, a DNS server must be available on the network to resolve the name to an IP address.

- The IP address of an intermediate router if the configuration server is on a different LAN segment from the new router.
2. After the new router acquires an IP address, the autoinstallation process on the router attempts to download a configuration file in the following ways:
 - a. If the configuration file is specified as a URL, the router fetches the configuration file from the URL by using HTTP, FTP, or TFTP, depending on the protocol specified in the URL.
 - b. If the DHCP server specifies the host-specific configuration file (boot file) *hostname.conf*, the router uses that filename in the TFTP server request. (In the filename, **hostname** is the hostname of the

new router.) The autoinstallation process on the new router makes three unicast TFTP requests for **hostname.conf**. If these attempts fail, the router broadcasts three requests to any available TFTP server for the file.

- c. If the new router cannot locate **hostname.conf**, the autoinstallation process unicasts or broadcasts TFTP requests for a default router configuration file called **network.conf**, which contains hostname-to-IP address mapping information, to attempt to find its hostname.
 - d. If **network.conf** contains no hostname entry for the new router, the autoinstallation process sends out a DNS request and attempts to resolve the new router's IP address to a hostname.
 - e. If the new router can determine its hostname, it sends a TFTP request for the **hostname.conf** file.
 - f. If the new router is unable to map its IP address to a hostname, it sends TFTP requests for the default configuration file **router.conf**.
3. After the new router locates a configuration file on a TFTP server, the autoinstallation process downloads the file, installs the file on the router, and commits the configuration.

In a Junos Node Unifier (JNU) group that contains an MX Series router as a controller that manages satellite devices, such as EX Series Ethernet Switches, QFX Series devices, and ACX Series Universal Metro Routers, the autoinstallation functionality is supported for the satellite devices. JNU has an autoinstallation mechanism that enables a satellite device to configure itself out-of-the-box with no manual intervention, using the configuration available either on the network or locally through a removable media, or using a combination of both. This autoinstallation method is also called the *zero-touch* facility.

The zero-touch configuration delivers the following benefits:

- The router can be sent from the warehouse to the deployment site without any preconfiguration steps.
- The procedure required to deploy the device at the cell site is simplified, resulting in reduced operational and administrative costs.
- You can roll out large numbers of these devices in a very short time.

The factory default setting is autoinstallation-enabled. After you make the first configuration to the router, you can do either of the following:

- A JNU factory default file, **jnu-factory.conf**, is present in the **/etc/config/** directory and contains the configuration to perform autoinstallation on satellite devices. The zero-touch configuration can be disabled by including the `delete-after-commit` statement at the `[edit system autoinstallation]` hierarchy level and committing the configuration. This way, the saved configuration is used the next time the system reboots.

- Alternatively, if the router must get the configuration from the server each time a system reboot occurs, the zero-touch configuration must not be changed (that is, you must not include the delete-after-commit statement at the [edit system autoinstallation] hierarchy level and commit the settings).

SEE ALSO

autoinstallation
delete-after-commit (JNU Satellites)

Change History Table

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

Release	Description
15.1X49-D60	Starting in Junos OS Release 15.1X49-D60 and in Junos OS Release 17.3R1, on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices, some of the factory-default configurations are changed.

Configuring Autoinstallation of Configuration Files (Junos OS)

IN THIS SECTION

- [Configuring Autoinstallation of Configuration Files \(CLI Procedure\) | 568](#)
- [Example: Configure Autoinstallation on SRX Series Devices | 570](#)
- [Verifying Autoinstallation Status | 574](#)
- [Autoinstalling a Configuration File from a Disk-on-Key USB Memory Stick onto an EX2200 or EX3300 Switch | 575](#)
- [Configuring Autoinstallation on JNU Satellite Devices | 578](#)
- [Verifying Autoinstallation on JNU Satellite Devices | 581](#)

Autoinstallation is an automated process and does not require any specific configuration on a device. To simplify the process, you can specify one or more interfaces, protocols, and configuration servers to be used for autoinstallation.

Configuring Autoinstallation of Configuration Files (CLI Procedure)

Autoinstallation is the automatic configuration of a device over the network from a pre-existing configuration file that you create and store on a configuration server. A configuration server is typically a Trivial File Transfer Protocol (TFTP) server. You can use autoinstallation to deploy multiple devices automatically from a central location in the network.

Before you can configure autoinstallation, you must enable autoinstallation to run when you power on a device already installed in your network. You enable it by specifying one or more interfaces, protocols, and configuration servers to be used for autoinstallation.

To enable autoinstallation to run, complete the following steps:

1. Ensure that a service such as Dynamic Host Configuration Protocol (DHCP) is available to assign an IP address to the device.
2. Configure a DHCP server on your network to meet your network requirements. You can configure a switch to operate as a DHCP server.
3. Create one of the following configuration files, and store it on a TFTP server (or HTTP server or FTP server) in the network:
 - A host-specific file with the name **hostname.conf** for each device undergoing autoinstallation. Replace **hostname** with the name of a device. The **hostname.conf** file typically contains all the configuration information necessary for the device with this hostname.
 - A default configuration file named **device.conf** with the minimum configuration necessary to enable you to telnet into the new device for further configuration.
4. Physically attach the device to the network using a Gigabit Ethernet port.
5. If you configured the DHCP server to provide only the TFTP server hostname, add an IP address-to-hostname mapping entry for the TFTP server. Map the TFTP server hostname to the DNS database file on the Domain Name System (DNS) server in the network.
6. If the device is not on the same network segment as the DHCP server (or other device providing IP address resolution), configure an existing device as an intermediate device to receive TFTP and DNS requests and forward them to the TFTP server and the DNS server. You must configure the LAN or serial interface on the intermediate device with the IP addresses of the hosts providing TFTP and DNS services. Connect this interface to the device.
7. If you are using **hostname.conf** files for autoinstallation, you must also complete the following tasks:
 - Configure the DHCP server to provide a **hostname.conf** filename to each device. Each device uses its **hostname.conf** filename to request a configuration file from the TFTP server. Copy the necessary **hostname.conf** configuration files to the TFTP server.

- Create a default configuration file named **network.conf**, and copy it to the TFTP server. This file contains IP-address-to-hostname mapping entries. If the DHCP server does not send a **hostname.conf** filename to a new device, the device uses **network.conf** to resolve its hostname based on its IP address.

Alternatively, you can add the IP-address-to-hostname mapping entry for the device to a DNS database file.

The device uses the hostname to request a **hostname.conf** file from the TFTP server.

Before you explicitly enable and configure autoinstallation on the device, perform these tasks as needed for your network configuration:

To configure autoinstallation:

1. Specify the URL address of one or more servers from which to obtain configuration files.

```
[edit system]
user@host# set autoinstallation configuration-servers tftp://tftpconfig.example.com
```



NOTE: You can also use an FTP address such as **ftp://*user.password@sftpconfig.example.com***.

2. Configure one or more Ethernet interfaces to perform autoinstallation and one or two procurement protocols for each interface. The switch uses the protocols to send a request for an IP address for the interface:

```
[edit system]
user@host# set autoinstallation interfaces ge-0/0/0 bootp
```

To verify autoinstallation, from the CLI enter the `show system autoinstallation status` command.

Example:

```
user@host> show system autoinstallation status
Autoinstallation status:
Master state: Active
Last committed file: None
Configuration server of last committed file: 10.25.100.1
Interface:
  Name: ge-0/0/0
  State: Configuration Acquisition
```

```

Acquired:
  Address: 192.168.124.75
  Hostname: host-ge-000
  Hostname source: DNS
  Configuration filename: device-ge-000.conf
  Configuration filename server: 10.25.100.3
Address acquisition:
  Protocol: DHCP Client
  Acquired address: None
  Protocol: RARP Client
  Acquired address: None
Interface:
  Name: ge-0/0/1
  State: None
Address acquisition:
  Protocol: DHCP Client
  Acquired address: None
  Protocol: RARP Client
  Acquired address: None

```

Example: Configure Autoinstallation on SRX Series Devices

IN THIS SECTION

- [Requirements | 570](#)
- [Overview | 571](#)
- [Configuration | 571](#)
- [Verification | 573](#)

This example shows how to configure a device for autoinstallation.

Requirements

Before you begin:

- Configure a DHCP server on your network to meet your network requirements. You can configure a device to operate as a DHCP server.
- Create one of the following configuration files, and store it on a TFTP server in the network:

- A host-specific file with the name *hostname.conf* for each device undergoing autoinstallation. Replace *hostname* with the name of a device. The *hostname.conf* file typically contains all the configuration information necessary for the device with this hostname.
- A default configuration file named *router.conf* with the minimum configuration necessary to enable you to telnet into the new device for further configuration.
- Physically attach the device to the network using one or more of the following interface types:
 - Fast Ethernet
 - Gigabit Ethernet
 - Serial with HDLC encapsulation

Overview

No configuration is required on a device on which you are performing autoinstallation, because it is an automated process. However, to simplify the process, you can specify one or more interfaces, protocols, and configuration servers to be used for autoinstallation.

The device uses these protocols to send a request for an IP address for the interface.

- BOOTP—Sends requests over all interfaces.
- RARP—Sends requests over Ethernet interfaces.

It is necessary to additionally configure the family *inet* under the interface using the `set interfaces ge-0/0/X unit 0 family inet` command for the SRX Series Firewall to send dhcp requests out.

Configuration

IN THIS SECTION

- [Procedure | 571](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy

and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set system autoinstallation configuration-servers tftp://tftpconfig.sp.com
set system autoinstallation interfaces ge-0/0/0 bootp rarp
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a device for autoinstallation:

1. Enable autoinstallation and specify the URL address of one or more servers from which to obtain configuration files.

```
[edit system]
user@host# set autoinstallation configuration-servers tftp://tftpconfig.sp.com
```

You can also use an FTP address, for example, `ftp://user:password@sftpconfig.sp.com`.

2. Configure one or more Ethernet or serial interfaces to perform autoinstallation, and configure one or two procurement protocols for each interface.

```
[edit system]
user@host# set autoinstallation interfaces ge-0/0/0 bootp rarp
```

Results

From configuration mode, confirm your configuration by entering the `show system autoinstallation status` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show system autoinstallation status
```

```
Autoinstallation status:
Master state: Active
```

```

Last committed file: None
Configuration server of last committed file: 10.25.100.1
Interface:
  Name: ge-0/0/0
  State: Configuration Acquisition
  Acquired:
    Address: 192.168.124.75
    Hostname: host-ge-000
    Hostname source: DNS
    Configuration filename: router-ge-000.conf
    Configuration filename server: 10.25.100.3
  Address acquisition:
    Protocol: BOOTP Client
    Acquired address: None
    Protocol: RARP Client
    Acquired address: None

```

If you are done configuring the device, enter `commit` from configuration mode.

When there is a user-specified configuration for a particular interface, delete the factory default for that interface. Having two configurations for the same device might lead to errors. For example, if PPP encapsulation is set on a T1 interface through user configuration while the factory default configuration configures CISCO HLDC on the same interface, then the interface might not come up and the following error is logged in the message file: **"DCD_CONFIG_WRITE_FAILED failed."**

Verification

IN THIS SECTION

- [Verifying Autoinstallation | 573](#)

Confirm that the configuration is working properly.

Verifying Autoinstallation

Purpose

Verify that the device has been configured for autoinstallation.

Action

From operational mode, enter the `show system autoinstallation status` command. The output shows the settings configured for autoinstallation. Verify that the values displayed are correct for the device when it is deployed on the network.

Verifying Autoinstallation Status

IN THIS SECTION

- [Purpose | 574](#)
- [Action | 574](#)
- [Meaning | 575](#)

Purpose

Display the status of the autoinstallation feature.

Action

From the CLI, enter the `show system autoinstallation status` command.

Sample Output

command-name

```
user@switch> show system autoinstallation status
Autoinstallation status:
Master state: Active
Last committed file: None
Configuration server of last committed file: 10.25.100.1
Interface:
  Name: ge-0/0/0
  State: Configuration Acquisition
  Acquired:
    Address: 192.168.124.75
    Hostname: host-ge-000
    Hostname source: DNS
```

```

Configuration filename: switch-ge-000.conf
Configuration filename server: 10.25.100.3
Address acquisition:
  Protocol: DHCP Client
  Acquired address: None
  Protocol: RARP Client
  Acquired address: None
Interface:
  Name: ge-0/0/1
  State: None
  Address acquisition:
    Protocol: DHCP Client
    Acquired address: None
    Protocol: RARP Client
    Acquired address: None

```

Meaning

The output shows the settings configured for autoinstallation. Verify that the values displayed are correct for the switch when it is deployed on the network.

Autoinstalling a Configuration File from a Disk-on-Key USB Memory Stick onto an EX2200 or EX3300 Switch

If you have a new EX2200 or EX3300 switch, you can use a Disk-on-Key USB memory stick (“USB key”) to configure the switch, using either a text configuration file or an XML configuration file.

Before you begin this task, ensure you have the following items:

- A management device (PC or laptop).
- A Junos Space platform to generate a valid XML file (if you will be installing the XML configuration file).
- A Disk-on-Key device with one of the following 16-bit or 32-bit FAT file systems:
 - DOS 3.0+ 16-bit FAT (up to 32 MB)
 - DOS 3.31+ 16-bit FAT (more than 32 MB)
 - FAT32
 - FAT32, LBA-mapped
 - 16-bit FAT, LBA-mapped

- An EX2200 or EX3300 switch with the factory configuration. If other Junos OS configuration files exist on the switch, the switch cannot read the **juniper-config.txt** or **juniper-config.xml** file from the Disk-on-Key device.



NOTE: The USB-based autoinstallation process overrides the network-based autoinstallation process. If the switch detects a Disk-on-Key device containing a valid configuration file during autoinstallation, it configures the switch by using the configuration file on the Disk-on-Key device instead of fetching the configuration from the network.

If both **juniper-config.txt** and **juniper-config.xml** files are on the Disk-on-Key device, the switch uses the text (txt) file.

To configure the switch by using a Disk-on-Key device that contains the configuration file in *text format*:

1. Using a text editor on the PC or laptop, create the configuration file, named **juniper-config.txt**, as a sequence of configuration commands (set commands). To reuse the configuration from another switch, save the configuration in configuration mode as a sequence of configuration commands on the switch using the `show | display set | save filename` command and then copying the file to the PC or switch as **juniper-config.txt**.



NOTE: Ensure that the first line in the **juniper-config.txt** is `edit` and that the last line in the file is `commit and-quit`.

2. Copy the **juniper-config.txt** file to the Disk-on-Key device.
3. Plug the Disk-on-Key device into the USB port on the switch.
4. Power on the switch.
5. Observe the LEDs on the Disk-on-Key device, and wait as the switch starts and then accesses the Disk-on-Key device.

The switch reads the **juniper-config.txt** file from the Disk-on-Key device and commits the configuration.



NOTE: Before you remove the Disk-on-Key device from the switch, ensure that the configuration has been applied to the switch. You can issue the `show configuration operational mode` command on the switch to see the configuration.

Then remove the Disk-on-Key device from the switch.

The configuration of the switch is complete.

To configure the switch by using a Disk-on-Key device that contains the configuration file in *XML format*:

1. Power on the switch.
2. Configure the switch to use autoinstallation:
 - a. Load the factory default configuration:

```
[edit]
user@switch# load factory-default
```

- b. Set the switch for autoinstallation:

```
[edit]
user@switch# set system autoinstallation delete-upon-commit
```

- c. Set the root authentication password:

```
[edit]
user@switch# set system root-authentication plain-text-password
```

- d. Commit the changes:

```
[edit]
user@switch# commit
```

3. Power off the switch.
4. Using the Junos Space platform, create a valid configuration file in XML format, and name it **juniper-config.xml**.
5. Copy the **juniper-config.xml** file to the Disk-on-Key device.
6. Plug the Disk-on-Key device into the USB port on the switch.
7. Power on the switch.
8. Observe the LEDs on the Disk-on-Key device, and wait as the switch starts and then accesses the Disk-on-Key device.

The switch reads the **juniper-config.xml** file from the Disk-on-Key device and commits the configuration.



NOTE: Before you remove the Disk-on-Key device from the switch, ensure that the configuration has been applied to the switch. You can issue the `show configuration operational mode` command on the switch to see the configuration.

Then remove the Disk-on-Key device from the switch.

The configuration of the switch is complete.

SEE ALSO

show system autoinstallation status

[Installing Software on EX Series Switches | 143](#)

Configuring Autoinstallation on JNU Satellite Devices

No configuration is required on a device on which you are performing autoinstallation because it is an automated process. However, to simplify the process, you can specify one or more interfaces, protocols, and configuration servers to be used for autoinstallation. In this scenario, satellite devices, such as EX Series Ethernet Switches, QFX Series devices, and ACX Series Universal Metro Routers, that are managed by the controller are considered.

To configure autoinstallation:

1. Load the JNU factory-default configuration file on the satellite device to enable the device to function in JNU mode.

```
user@satellite# load override /etc/config/jnu-factory.conf
```

An override operation discards the current candidate configuration and loads the configuration in the specified filename or the one that you type at the terminal. When you use the override option and commit the configuration, all system processes reparse the configuration.

2. Specify the URL address of one or more servers from which to obtain configuration files:

```
[edit system]
user@host# set autoinstallation configuration-servers tftp://tftpconfig.sp.com
```



NOTE: You can also use an HTTP or FTP address—for example, **http://*user.password*@httpconfig.sp.com** or **ftp://*user.password*@sftpconfig.sp.com**.

3. Configure one or more Ethernet interfaces to perform autoinstallation and IP address acquisition protocols for each interface. The router uses the protocols to send a request for an IP address for the interface:

```
[edit system]
user@host# set autoinstallation interfaces ge-0/0/0 bootp
```

4. Set the root password, entering a clear-text password that the system will encrypt, a password that is already encrypted, or an SSH public key string.

Choose one of the following:

- To enter a clear-text password, use the following command:

```
[edit system]
user@host# set root-authentication plain-text-password
New password: type password here
Retype new password: retype password here
```

- To enter a password that is already encrypted, use the following command:

```
[edit]
root# set system root-authentication encrypted-password encrypted-password
```

- To enter an SSH public key, use the following command:

```
[edit]
root# set system root-authentication ssh-rsa key
```

5. Save the Junos OS configuration changes, activate the configuration on the device, and exit configuration mode, using the `commit-and-quit` command.

```
[edit]
user@host# commit-and-quit
```

When the satellite device reboots, it triggers the autoinstallation mechanism to retrieve its initial configuration and downloads the settings from the configuration file stored on a configuration server in the network. On the controller, you must enable the FTP service by using the `set system services ftp` command and save the configuration on the satellite device at the `/var/jnu/` directory.

The following configuration is generated on the satellite device as a result of the preceding procedure to configure autoinstallation:

```
system {
  autoinstallation {
    traceoptions {
      flags {
        all;
      }
      file autod;
      level all;
    }
    delete-after-commit; /* After initial config, no need to keep */
    interfaces {
      ge-* {
        bootp;
      }
      xe-* {
        bootp;
      }
      configuration-servers {
        "ftp://192.168.0.1/var/jnu/sat1.conf";
      }
    }
  }
  root-authentication {
    encrypted-password "$ABC123";
  }
}
```

SEE ALSO

autoinstallation

delete-after-commit (JNU Satellites)

Verifying Autoinstallation on JNU Satellite Devices

IN THIS SECTION

- Purpose | 581
- Action | 581
- Meaning | 582

Purpose

After you have configured autoinstallation, display the status of autoinstallation on a satellite device, such as an ACX Series router, an EX Series switch, or a QFX Series device, in a Junos Node Unifier (JNU) group that is managed by a controller, which is an MX Series router.

Action

From the CLI, enter the `show system autoinstallation status` command. The following example displays the autoinstallation settings of an ACX Series router that operates as a satellite in a JNU group.

Sample Output

command-name

```
user@host> show system autoinstallation status
Autoinstallation status:
  Master state: Active
  Last committed file: None
  Configuration server of last committed file: 10.25.100.1
  Interface:
    Name: ge-0/1/0
    State: Configuration Acquisition
  Acquired:
    Address: 192.168.124.75
    Hostname: host-ge-000
    Hostname source: DNS
    Configuration filename: router-ge-000.conf
```

```

Configuration filename server: 10.25.100.3
Address acquisition:
  Protocol: DHCP Client
  Acquired address: None
  Protocol: RARP Client
  Acquired address: None
Interface:
  Name: ge-0/1/1
  State: None
  Address acquisition:
    Protocol: DHCP Client
    Acquired address: None
    Protocol: RARP Client
    Acquired address: None

```

Meaning

The output shows the settings configured for autoinstallation. Verify that the values displayed are correct for the router when it is deployed on the network.

SEE ALSO

autoinstallation

delete-after-commit (JNU Satellites)

show system autoinstallation status

Configuring Autoinstallation of Configuration Files on ACX Series (Junos OS)

IN THIS SECTION

- [ACX Series Autoinstallation Overview | 583](#)
- [Before You Begin Autoinstallation on an ACX Series Universal Metro Router | 585](#)
- [Autoinstallation Configuration of ACX Series Universal Metro Routers | 586](#)

- [Verifying Autoinstallation on ACX Series Universal Metro Routers | 587](#)
- [USB Autoinstallation on ACX Series Routers | 588](#)
- [Autoinstallation on ACX Series Routers in Hybrid Mode Overview | 589](#)
- [Prerequisites for Autoinstallation on ACX Series Routers in Hybrid Mode | 590](#)
- [Autoinstallation Process on a New ACX Series Router in Hybrid Mode | 591](#)
- [Configuring Autoinstallation of ACX Series Routers in Hybrid Mode | 594](#)

Autoinstallation is an automated process and does not require any specific configuration on a device. To simplify the process, you can specify one or more interfaces, protocols, and configuration servers to be used for autoinstallation.

ACX Series Autoinstallation Overview

IN THIS SECTION

- [Supported Autoinstallation Interfaces and Protocols | 584](#)
- [Typical Autoinstallation Process on a New Router | 584](#)

Autoinstallation provides automatic configuration for a new router that you connect to the network and turn on, or for a router configured for autoinstallation. The autoinstallation process begins anytime a router is powered on and cannot locate a valid configuration file in the CompactFlash (CF) card. Typically, a configuration file is unavailable when a router is powered on for the first time, or if the configuration file is deleted from the CF card. The autoinstallation feature enables you to deploy multiple routers from a central location in the network.

For the autoinstallation process to work, you must store one or more host-specific or default configuration files on a configuration server in the network and have a service available—typically Dynamic Host Configuration Protocol (DHCP)—to assign an IP address to the router.

Autoinstallation takes place automatically when you connect an Ethernet on a new Juniper Networks router to the network and power on the router. To simplify the process, you can explicitly enable autoinstallation on a router and specify a configuration server, an autoinstallation interface, and a protocol for IP address acquisition.

Supported Autoinstallation Interfaces and Protocols

Before autoinstallation on a router can take place, the router must acquire an IP address or a USB key. The protocol or protocols you choose for IP address acquisition determine the router interface to connect to the network for autoinstallation. The router detects the connected interface and requests an IP address with a protocol appropriate for the interface. Autoinstallation is supported over an Ethernet LAN interface. For IP address acquisition, the ACX Series router uses DHCP, BOOTP, or Reverse Address Resolution Protocol (RARP) on an Ethernet LAN interface.

If the server with the autoinstallation configuration file is not on the same LAN segment as the new router, or if a specific router is required by the network, you must configure an intermediate router directly attached to the new router, through which the new router can send HTTP, FTP, Trivial File Transfer Protocol (TFTP), BOOTP, and Domain Name System (DNS) requests. In this case, you specify the IP address of the intermediate router as the location to receive HTTP, FTP, or TFTP requests for autoinstallation.

Typical Autoinstallation Process on a New Router

When a router is powered on for the first time, it performs the following autoinstallation tasks:

1. The new router sends out DHCP, BOOTP, or RARP requests on each connected interface simultaneously to obtain an IP address.

If a DHCP server responds, it provides the router with some or all of the following information:

- An IP address and subnet mask for the autoinstallation interface.
- The location of the TFTP (typically), Hypertext Transfer Protocol (HTTP), or FTP server on which the configuration file is stored.
- The name of the configuration file to be requested from the HTTP, FTP, or TFTP server.
- The IP address or hostname of the HTTP, FTP, or TFTP server.

If the DHCP server provides only the hostname, a DNS server must be available on the network to resolve the name to an IP address.

- The IP address of an intermediate router if the configuration server is on a different LAN segment from the new router.
2. After the new router acquires an IP address, the autoinstallation process on the router attempts to download a configuration file in the following ways:
 - a. If the configuration file is specified as a URL, the router fetches the configuration file from the URL by using HTTP, FTP, or TFTP depending on the protocol specified in the URL.

- b. If the DHCP server specifies the host-specific configuration file (boot file) **hostname.conf**, the router uses that filename in the TFTP server request. (In the filename, **hostname** is the hostname of the new router.) The autoinstallation process on the new router makes three unicast TFTP requests for **hostname.conf**. If these attempts fail, the router broadcasts three requests to any available TFTP server for the file.
 - c. If the new router cannot locate **hostname.conf**, the autoinstallation process unicasts or broadcasts TFTP requests for a default router configuration file called **network.conf**, which contains hostname-to-IP address mapping information, to attempt to find its hostname.
 - d. If **network.conf** contains no hostname entry for the new router, the autoinstallation process sends out a DNS request and attempts to resolve the new router's IP address to a hostname.
 - e. If the new router can determine its hostname, it sends a TFTP request for the **hostname.conf** file.
 - f. If the new router is unable to map its IP address to a hostname, it sends TFTP requests for the default configuration file **router.conf**.
3. After the new router locates a configuration file on a TFTP server, autoinstallation downloads the file, installs the file on the router, and commits the configuration.

Before You Begin Autoinstallation on an ACX Series Universal Metro Router

To configure a router for autoinstallation, complete the following tasks:

- Make sure you have a DHCP server on your network to meet your network requirements.
- Create one of the following configuration files and store it on an HTTP, FTP, or TFTP server in the network:
 - A host-specific file with the name **hostname.conf** for each router undergoing autoinstallation. Replace **hostname** with the name of a router. The **hostname.conf** file typically contains all the configuration information necessary for the router with this hostname.
 - A default configuration file named **router.conf** with the minimum configuration necessary to enable you to telnet into the new router for further configuration.
- Physically attach the router to the network using a Gigabit Ethernet interface.
- If you configure the DHCP server to provide only the HTTP, FTP, or TFTP server hostname, add an IP address-to-hostname mapping entry for the HTTP, FTP, or TFTP server to the DNS database file on the DNS server in the network.
- If the new router is not on the same network segment as the DHCP server (or other router providing IP address resolution), configure an existing router as an intermediate to receive HTTP, FTP, or TFTP and DNS requests and forward them to the HTTP, FTP, or TFTP and DNS servers. You must

configure the LAN on the intermediate router with the IP addresses of the hosts providing HTTP, FTP, or TFTP and DNS service. Connect this interface to the new router.

- If you are using **hostname.conf** files for autoinstallation of host-specific configuration files, you must also complete the following tasks:
 - Configure the DHCP server to provide a **hostname.conf** filename to each new router. Each router uses its **hostname.conf** filename to request a configuration file from the TFTP server. Copy the necessary **hostname.conf** configuration files to the TFTP server.
 - Create a default configuration file named **network.conf** and copy it to the TFTP server. This file contains IP address-to-hostname mapping entries. If the DHCP server does not send a **hostname.conf** filename to a new router, the router uses **network.conf** to resolve its hostname based on its IP address.

Alternatively, you can add the IP address-to-hostname mapping entry for the new router to a DNS database file.

The router uses the hostname to request a **hostname.conf** file from the server.

Autoinstallation Configuration of ACX Series Universal Metro Routers

No configuration is required on a router on which you are performing autoinstallation because it is an automated process. However, to simplify the process, you can specify one or more interfaces, protocols, and configuration servers to be used for autoinstallation.

To configure autoinstallation:

1. Specify the URL address of one or more servers from which to obtain configuration files.

```
[edit system]
user@host# set autoinstallation configuration-servers tftp://tftpconfig.sp.com
```



NOTE: You can also use an HTTP or FTP address—for example, **http://user.password@httpconfig.sp.com** or **ftp://user.password@sftpconfig.sp.com**.

2. Configure one or more Ethernet interfaces to perform autoinstallation and IP address acquisition protocols for each interface. The router uses the protocols to send a request for an IP address for the interface:

```
[edit system]
user@host# set autoinstallation interfaces ge-0/0/0 bootp
```

Verifying Autoinstallation on ACX Series Universal Metro Routers

IN THIS SECTION

- Purpose | 587
- Action | 587
- Meaning | 588

Purpose

After you have configured autoinstallation, display the status of autoinstallation on an ACX Series router.

Action

From the CLI, enter the `show system autoinstallation status` command.

Sample Output

`show system autoinstallation status`

```
user@host> show system autoinstallation status
Autoinstallation status:
  Master state: Active
  Last committed file: None
  Configuration server of last committed file: 10.25.100.1
  Interface:
    Name: ge-0/1/0
    State: Configuration Acquisition
    Acquired:
      Address: 192.168.124.75
      Hostname: host-ge-000
      Hostname source: DNS
      Configuration filename: router-ge-000.conf
      Configuration filename server: 10.25.100.3
  Address acquisition:
    Protocol: DHCP Client
    Acquired address: None
```

```

    Protocol: RARP Client
    Acquired address: None
Interface:
    Name: ge-0/1/1
    State: None
    Address acquisition:
        Protocol: DHCP Client
        Acquired address: None
        Protocol: RARP Client
        Acquired address: None

```

Meaning

The output shows the settings configured for autoinstallation. Verify that the values displayed are correct for the router when it is deployed on the network.

SEE ALSO

autoinstallation

show system autoinstallation status

USB Autoinstallation on ACX Series Routers

If you have a new ACX Series router, you can use a Disk-on-Key USB memory stick (“USB key”) to configure the router.

This configuration method has the following requirements:

- A management device (PC or laptop).
- A Disk-on-Key device with one of the following 16-bit or 32-bit file allocation table (FAT) file systems:
 - DOS 3.0+ 16-bit FAT (up to 32 MB)
 - DOS 3.31+ 16-bit FAT (over 32 MB)
 - FAT32
 - FAT32, LBA-mapped
 - 16-bit FAT, LBA-mapped

- An ACX Series router with the factory configuration. If other Junos OS configuration files exist on the router, the router cannot read the **juniper-config.txt** file from the Disk-on-Key device.



NOTE: The USB-based autoinstallation process overrides the network-based autoinstallation process. If the ACX Series router detects a USB Disk-on-Key device containing a valid configuration file during autoinstallation, it configures the router using the configuration file on Disk-on-Key instead of fetching the configuration from the network.

To configure an ACX Series router using Disk-on-Key:

1. Using a text editor on a PC or laptop, create the configuration file, named *juniper-config.txt*, as a sequence of configuration commands ("set" commands). To reuse configuration from another ACX Series router, the configuration can be saved in configuration mode as a sequence of configuration commands on the router using the "show | display set | save <filename>" command and then copying the <filename> to the PC or router as *juniper-config.txt*.
2. Copy the *juniper-config.txt* file to a Disk-on-Key device.
3. Plug the Disk-on-Key device into the USB port on the new ACX Series router.
4. Power on the router by pressing the POWER button on the front panel. Wait for the router to start and access the Disk-on-Key device (observe the LEDs on the Disk-on-Key device).
The router reads the *juniper-config.txt* file from the Disk-on-Key device and commits the configuration.
5. Remove the Disk-on-Key device from the router.
6. The configuration of the router is complete.

Autoinstallation on ACX Series Routers in Hybrid Mode Overview

The ACX Series router has an autoinstallation mechanism that allows the router to configure itself out-of-the-box with no manual intervention, using the configuration available either on the network, locally through a removable media, or a combination of both.

Autoinstallation process delivers the following benefits:

- The router can be sent from the warehouse to the deployment site without any pre-configuration steps.
- The procedure required to deploy the device at the cell site is simplified, resulting in reduced operational and administrative costs.
- You can roll out large numbers of these devices in a very short time.

ACX Series routers support the retrieval of partial configuration from an external USB storage device plugged into the router's USB port during the autoinstallation process. This partial configuration in turn

facilitates the network mode of autoinstallation to retrieve the complete configuration file from the network. This method is called hybrid mode of autoinstallation.

Autoinstallation process operates in three modes:

- **USB mode**—Autoinstallation obtains the required configuration from the configuration file saved in an external USB storage device plugged into the router.
- **Network Mode**—Autoinstallation triggers IP address acquisition mechanism (the router sends out DHCP or RARP requests on each connected interface simultaneously) to obtain an IP address. Once the router has an IP address, it sends a request to the specified configuration server and downloads and installs the configuration.
- **Hybrid mode**—Autoinstallation obtains partial configuration from an external USB storage device and uses that configuration to obtain the complete configuration file in network mode. This mode is a combination of USB mode and Network mode.

On the different ACX Series routers, autoinstallation is supported on the following Gigabit Ethernet (ge) and 10- Gigabit Ethernet (xe) interfaces:

- On ACX1000 routers, interfaces ge-0/1/0 through ge-0/1/7, and ge-0/2/0 through ge-0/2/3
- On ACX1100 routers, interfaces ge-0/0/0 through ge-0/0/7, and ge-0/1/0 through ge-0/1/3
- On ACX2000 routers, interfaces ge-0/1/0 through ge-0/1/7, ge-0/2/0 through ge-0/2/1, and xe-0/3/0 through xe-0/3/1
- On ACX2100 routers, interfaces ge-1/0/0 through ge-1/0/3, ge-1/1/0 through ge-1/1/3, ge-1/2/0 through ge-1/2/1, and xe-1/3/0 through xe-1/3/1
- On ACX2200 routers, interfaces ge-0/0/0 through ge-0/0/3, ge-0/1/0 through ge-0/1/3, ge-0/2/0 through ge-0/2/1, and xe-0/3/0 through xe-0/3/1
- On ACX4000 routers, interfaces ge-0/0/0 through ge-0/0/7, ge-0/1/0 through ge-0/1/1, ge-1/0/0 through ge-1/0/5, ge-1/1/0 through ge-1/1/5 , and xe-0/2/0 through xe-0/2/1

Prerequisites for Autoinstallation on ACX Series Routers in Hybrid Mode

Before you perform autoinstallation on a router in hybrid mode, complete the following tasks:

Using a text editor on a PC or laptop, create the configuration file, named *juniper-config.txt*, as a sequence of configuration commands ("set" commands). To reuse configuration from another ACX Series router, the configuration can be saved in configuration mode as a sequence of configuration commands on the router using the "show | display set | save <filename>" command and then copying the <filename> to the PC or router as *juniper-config.txt*.

You must copy the *juniper-config.txt* file to an external USB storage device. Plug the USB device into the USB port on the new ACX Series router. When you power on the router, the router first attempts to

access the external USB storage device. The router reads the *juniper-config.txt* file from the external USB storage device and commits the configuration.



NOTE: For autoinstallation process to switch to the network mode, the `continue-network-mode` statement must be present in the autoinstallation stanza at the `[edit system autoinstallation]` hierarchy level of the **juniper-config.txt** configuration file. The presence of the `continue-network-mode` statement in the **juniper-config.txt** file causes the router to consider it as a partial configuration. Otherwise, if the `continue-network-mode` statement is not present in the **juniper-config.txt** file, the router considers the configuration on the external USB storage device as the complete configuration and it will not switch to the network mode.

Perform all of the steps described in the ["Before You Begin Autoinstallation on an ACX Series Universal Metro Router"](#) on page 585 section, which prepares the router for network-based autoinstallation.

Autoinstallation Process on a New ACX Series Router in Hybrid Mode

You can perform autoinstallation on a new ACX Series router in hybrid mode, which is a combination of the USB-based autoinstallation process and the network-based autoinstallation process.

This configuration method has the following requirements:

- A management device (PC or laptop).
- An external USB storage device with one of the following 16-bit or 32-bit file allocation table (FAT) file systems:
 - DOS 3.0+ 16-bit FAT (up to 32 MB)
 - DOS 3.31+ 16-bit FAT (over 32 MB)
 - FAT32
 - FAT32, LBA-mapped
 - 16-bit FAT, LBA-mapped

BOOTP, RARP and DHCP are the supported protocols for acquisition of IP address of the router and TFTP, FTP, and HTTP are the supported protocols for downloading the configuration file from an external server URL on which the configuration file is stored.

The following operations occur during autoinstallation in hybrid mode on ACX Series routers:

1. When a new ACX Series router is powered on for the first time, the router performs the following autoinstallation tasks: The router boots the Junos OS image. The management process (mgd) is invoked and it determines whether a valid configuration exists on the router's Flash memory. If a

valid configuration is not present on the router, it loads and commits the factory-default configuration.

2. If the factory-default configuration contains the `autoinstallation` configuration stanza at the `[edit system]` hierarchy level, the autoinstallation process is triggered.
3. The autoinstallation process detects whether an external USB storage device is connected to the router and examines whether the USB device contains a valid configuration file. If the USB storage device contains a configuration file named **juniper-config.txt**, the router reads the **juniper-config.txt** file and commits the configuration.
4. If the **juniper-config.txt** file on the external USB storage device contains `continue-network-mode` statement, the configuration is treated as partial configuration. The autoinstallation process uses this partial configuration to obtain the complete configuration file from a server on the network. At this stage, the router completes the USB mode of the autoinstallation procedure and switches to the network mode of the autoinstallation procedure.



NOTE: The `continue-network-mode` statement must be present in the autoinstallation stanza at the `[edit system autoinstallation]` hierarchy level of the **juniper-config.txt** file.

5. After acquiring the partial configuration from the **juniper-config.txt** file, the configuration discovery procedure is initiated. For all physical Ethernet interfaces that transition to the up state, the autoinstallation process verifies whether autoinstallation is configured on that Ethernet interface. The autoinstallation process starts IP address acquisition mechanism to obtain IP address of the server followed by the configuration file retrieval mechanism.
6. For the interfaces that take part in the autoinstallation process, the IPv4 address discovery procedure is triggered. The new ACX Series router sends out DHCP, or BOOTP, or RARP requests on each connected interface simultaneously to obtain an IP address. The interfaces statement in the autoinstallation configuration stanza at the `[edit system]` hierarchy level in the factory-default configuration also specify the protocols to be used for IPv4 address discovery. If the interfaces statement is not configured, all the applicable protocols for an interface are used to send out requests on each connected Ethernet interface.
7. If an IPv4 address cannot be retrieved, the autoinstallation process starts the DHCP server on all participating interfaces (assigns static IP address in the form of 192.168.x.1 to allow a management station to connect to the router for manual configuration) and terminates the autoinstallation procedure.
8. If a DHCP server responds, it provides the router with some or all of the following information:
 - An IP address and subnet mask for the autoinstallation interface.
 - The location of the TFTP server on which the configuration file is stored.

- The name of the configuration file to be requested from the TFTP server.
- The IP address or hostname of the TFTP server.
- If the DHCP server provides configuration server hostname, a DNS server must be available on the network to resolve the name to an IP address.
- The IP address of an intermediate router if the configuration server is on a different LAN segment from the new router.



NOTE: To use HTTP or FTP server, you need to specify the URL of the configuration server under the [edit system autoinstallation configuration-servers] hierarchy level.

9. After an IPv4 address is retrieved for an interface, the interface is configured with that address and the autoinstallation process starts the configuration file discovery procedure. The autoinstallation process on the router attempts to download a configuration file in the following methods:
 - a. If the configuration file is specified as a URL, the router fetches the configuration file from the URL by using HTTP, FTP, or TFTP depending on the protocol specified in the URL.
 - b. If the DHCP server specifies the host-specific configuration file (either through file field option or boot file option or host name), the router uses that filename in the TFTP server request. In case of host name, the configuration filename is hostname.conf. The autoinstallation process on the new router makes unicast TFTP request for hostname.conf. If this attempt fails, the router broadcasts the request to any available TFTP server for the configuration file.
 - c. If the new router is unable locate the configuration file, the autoinstallation process unicasts or broadcasts TFTP requests for a default router configuration file called network.conf, which contains hostname-to-IP address mapping information, to attempt to find its hostname.
 - d. If network.conf contains no hostname entry for the new router, the autoinstallation process sends out a DNS request and attempts to resolve the new router's IP address to a hostname.
 - e. If the new router can determine its hostname, it sends a TFTP request for the hostname.conf file.
 - f. If the new router is unable to map its IP address to a hostname, it sends TFTP requests for the default configuration file router.conf.



NOTE: The autoinstallation process makes a maximum of three attempts to retrieve the configuration file by repeating the methods listed above (b to f). In case the autoinstallation process fails to retrieve the configuration file after three attempts, the autoinstallation process goes to start state.

- g. After the new router locates a configuration file on a TFTP server, autoinstallation downloads the file, installs the file on the router, and commits the configuration.

Configuring Autoinstallation of ACX Series Routers in Hybrid Mode

To configure the router for autoinstallation in hybrid mode, perform the following tasks:

Create a configuration file as *juniper-config.txt*.

1. Using a text editor on a PC or laptop, create the configuration file, named *juniper-config.txt*. This configuration file must contain a sequence of configuration commands ("set" commands).



NOTE: To reuse a configuration from another ACX Series router, save the configuration in configuration mode as a sequence of configuration commands on the router using the "show | display set | save <filename>" command and then copying the <filename> to the PC or router as *juniper-config.txt*.

2. Include the `continue-network-mode` statement at the [edit system autoinstallation] hierarchy level in the *juniper-config.txt* configuration file. The presence of the `continue-network-mode` statement causes the router to consider it as a partial configuration and the autoinstallation process switches to network mode to retrieve the complete configuration from a network server.

```
[edit system]
user@host# set autoinstallation continue-network-mode
```

3. Specify the URL address of one or more network servers from which to obtain the complete configuration.

```
[edit system]
user@host# set autoinstallation configuration-servers tftp://
username:password@tftpconfig.sp.com/filename.conf
```



NOTE: You can also use an HTTP or FTP address—for example, `http://user.password@httpconfig.sp.com/filename.conf` or `ftp://user.password@sftpconfig.sp.com/filename.conf`.

4. Specify the root authentication password.

```
[edit system]
user@host# set root-authentication encrypted-password "password"
```

5. Configure one or more Ethernet interfaces to perform autoinstallation and IP address acquisition protocols for each interface. The router uses the protocols to send a request for an IP address for the interface:

```
[edit system]
user@host# set autoinstallation interfaces ge-0/0/0 bootp
```



NOTE: Configuring an interface is optional. If an interface is configured, then autoinstallation process is triggered on the configured interface only. If an interface is not configured, then autoinstallation process is triggered on all the interfaces that are physically in link up state.

6. Copy the *juniper-config.txt* file to an external USB storage device.
7. Plug the external USB storage device to the router's USB port.

From configuration mode, confirm your configuration by entering the `show system autoinstallation status` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host> show system autoinstallation status
```

```
Autoinstallation status:
Master state: Active
Last committed file: None
Configuration server of last committed file: 10.25.100.1
Interface:
  Name: ge-0/0/0
  State: Configuration Acquisition
  Acquired:
    Address: 192.168.124.75
    Hostname: host-ge-000
    Hostname source: DNS
    Configuration filename: router-ge-000.conf
    Configuration filename server: 10.25.100.3
  Address acquisition:
    Protocol: BOOTP Client
    Acquired address: None
```

Protocol: RARP Client
Acquired address: None

SEE ALSO

autoinstallation

show system autoinstallation status

10

PART

Configuration Statements and Operational Commands

- [Configuration Statements and Operational Commands | 598](#)
-

Configuration Statements and Operational Commands

IN THIS CHAPTER

- [Junos CLI Reference Overview](#) | 598

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- [Junos CLI Reference](#)

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- [Configuration Statements](#)
- [Operational Commands](#)