



Introducing Junos OS Evolved

Release

18.3R1



Modified: 2019-01-30

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.

Introducing Junos OS Evolved

18.3R1

Copyright © 2019 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

Chapter 1	Overview of Junos OS Evolved	9
	Why Use Junos OS Evolved	9
	Benefits	9
	Resiliency (High Availability)	9
	Faster Innovation	10
	How Junos OS Evolved Differs from Junos OS	10
	CLI Statements Introduced in Junos OS Evolved	11
	CLI Statements Modified in Junos OS Evolved	11
	CLI Statements Deprecated in Junos OS Evolved	12
	CLI Commands Introduced in Junos OS Evolved	12
	CLI Commands Modified in Junos OS Evolved	14
	Changed Command Output	15
	Where to Find Information on Common Procedures	17
Chapter 2	Running Junos OS Evolved on a Device	19
	Booting Junos OS Evolved by Using a Bootable USB Drive	19
	Create a Bootable USB Drive Using a Windows Machine	19
	Create a Bootable USB Drive Using a MAC OS X	21
	Create a Bootable USB Drive Using a Switch or Router Running Junos OS Evolved	22
	Boot Junos OS Evolved on the QFX5200 Switch	23
	Recover Junos OS Evolved Using USB Scratch Install	24

List of Figures

Chapter 1	Overview of Junos OS Evolved	9
	Figure 1: Publish-Subscribe Model	10

List of Tables

Chapter 1	Overview of Junos OS Evolved	9
	Table 1: CLI Statement Introduced in Junos OS Evolved	11
	Table 2: CLI Statements Modified in Junos OS Evolved	12
	Table 3: CLI Statements Deprecated in Junos OS Evolved	12
	Table 4: CLI Commands Introduced in Junos OS Evolved	12
	Table 5: CLI Commands Modified in Junos OS Evolved	14
	Table 6: Changed Command Output	16

CHAPTER 1

Overview of Junos OS Evolved

- [Why Use Junos OS Evolved on page 9](#)
- [How Junos OS Evolved Differs from Junos OS on page 10](#)
- [Where to Find Information on Common Procedures on page 17](#)

Why Use Junos OS Evolved

Junos OS Evolved is the next generation of Junos operating system (OS). It is used just like Junos OS—the same CLI user interface, some of the same processes, and the same management and automation tools—but its infrastructure is entirely modernized, giving customers the resiliency, portability, faster innovation, and simplified upgrades they need.

Benefits

Junos OS Evolved runs natively on Linux, giving it direct access to all the Linux utilities and operations. Linux is a proven OS base that is widely used and familiar to programmers and operators alike. Junos OS Evolved is designed to be modular, allowing for compatible components that are changed to be restarted without requiring a reboot of the system. Junos OS Evolved is easily portable. Minimal work is required to make it work on any platform.

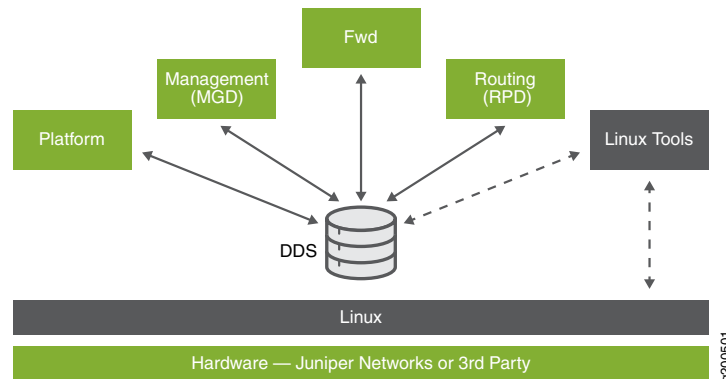
Resiliency (High Availability)

Junos OS Evolved minimizes changes to Linux, reducing churn in the base OS, thus contributing to the stability of the system.

A central database called the Data Distribution Service (DDS) holds all state. State is the retained information or status about each component that is preserved, shared across the system, and supplied during restarts. State includes both operational and configuration state, including committed configuration, and interface, routes, and hardware state.

The DDS does not interpret state. It works on the publish-subscribe model of application development (see [Figure 1 on page 10](#)). The sole job of the DDS is to hold state received from publishers and propagate state to consumers and subscribers. Each application publishes state to and subscribes to state from the DDS directly, making applications independent of each other.

Figure 1: Publish-Subscribe Model



Features dealing with legacy technology, such as ATM and Frame Relay, have been retired. Therefore, any impact from unused features is avoided.

Faster Innovation

Data and hardware are also formally modeled, leading to improved portability. You can quickly add new capabilities to the system. For example, if the capabilities of a card are already supported, then all that is needed is a manifest for that card specifying its capabilities.

Because all the state is formally modeled, you can autogenerate bindings to modern languages, such as Python, Scala, or C++. Modeling provides a common, known format, making it easy to convert to the schema for other modeling languages (such as OpenConfig or Thrift).

In Junos OS Evolved, modeled state is uniformly accessible, providing deeper visibility into network operations. Telemetry is streamed from devices to management systems automatically and continuously, and operators can subscribe in real time to the specific data they need using common data models. Such data is ready to be analyzed to identify trends and patterns and help with network automation, traffic optimization, and preventive troubleshooting.

Because it is Linux-compliant, Junos OS Evolved gives users access to Linux tools and operations that are already familiar to operators. Wherever possible, open-source tools and APIs are reused.

How Junos OS Evolved Differs from Junos OS

Junos OS Evolved mimics the functionality of Junos OS, but the way this functionality is achieved is different. There are, therefore, some differences in behavior. These differences are indicated throughout the Junos OS documentation. However, this section outlines the differences in one place, for your convenience.

- [CLI Statements Introduced in Junos OS Evolved on page 11](#)
- [CLI Statements Modified in Junos OS Evolved on page 11](#)

- [CLI Statements Deprecated in Junos OS Evolved on page 12](#)
- [CLI Commands Introduced in Junos OS Evolved on page 12](#)
- [CLI Commands Modified in Junos OS Evolved on page 14](#)
- [Changed Command Output on page 15](#)

CLI Statements Introduced in Junos OS Evolved

There is a new hierarchy level of statements in Junos OS Evolved that are not in classic Junos OS: **[edit security host-vpn]**. For more on these new statements, see [Table 1 on page 11](#).

Table 1: CLI Statement Introduced in Junos OS Evolved

Statement	Description	Link
[edit security host-vpn]	Configure a host-to-host VPN type of IPsec connection. Use the connections , ike-log , and ike-secrets statements at the [edit security host-vpn] hierarchy level to configure IKE and IPsec values.	<i>host-vpn</i>
[edit security host-vpn connections]	You can configure the additional algorithms aes256-sha384-modp3072 and aes256-gcm128-modp3072 at each of the following hierarchy levels: <ul style="list-style-type: none"> • [edit security host-vpn connections parent-connection-name ike-proposal] • [edit security host-vpn connections parent-connection-name children child-connection-name esp-proposal] 	<i>connections (Host VPN) and children</i>
[edit security host-vpn connections children child-name]	Statements at this hierarchy level include local-traffic-selector , remote , and remote-traffic-selector .	<i>children</i>
[edit security host-vpn connections dpd-delay]	Statement to support dead peer detection. The dead peer detection delay sends keepalives to know if a peer has gone dead.	<i>connections (Host VPN)</i>
[edit security host-vpn ike-log]	Statements at the [edit security host-vpn] hierarchy level used to configure IKE and IPsec values.	<i>ike-log</i>
[edit security host-vpn ike-secrets]	Statements at the [edit security host-vpn] hierarchy level used to configure IKE and IPsec values.	<i>ike-secrets</i>
[edit security host-vpn local]	Configure identity details for authenticating the local device during IKE negotiations.	<i>local (Host VPN)</i>
[edit security host-vpn remote]	Configure identity details for authenticating the remote device during IKE negotiations.	<i>remote (Host VPN)</i>

CLI Statements Modified in Junos OS Evolved

Some CLI statements have options that are changed in Junos OS Evolved. The parameters might change; the options might be different. See [Table 2 on page 12](#).

Table 2: CLI Statements Modified in Junos OS Evolved

Statement	Changed in Junos OS Evolved	Link
[edit interfaces <i>interface-name</i> unit <i>logical-unit-number</i> <i>vlan-id</i>]	The option vlan-id 0 is not supported for this statement.	<i>vlan-id (VLAN ID to Be Bound to a Logical Interface)</i>
[edit policy-options policy-statement <i>policy-name</i> term <i>term-name</i> then]	The following options are not supported in Junos OS Evolved 18.3R1: destination-class and source-class .	<i>Actions in Routing Policy Terms</i>
[edit system host-name]	When you specify the host-name statement, the Routing Engine name where the configuration is committed is appended to the hostname you specify.	<i>host-name</i>
[edit system login password]	The format option for this statement is limited to the following options: (md5 sha256 sha512).	<i>format (System Login)</i>
[edit system syslog host]	The syslog host is connected through the management interface. You cannot configure the source-address option if you configure the management-interface option.	<i>host (System)</i>

CLI Statements Deprecated in Junos OS Evolved

Some statements that are used in Junos OS are not used (deprecated) in Junos OS Evolved. For a listing of which CLI statements are deprecated in Junos OS Evolved, see [Table 3 on page 12](#).

Table 3: CLI Statements Deprecated in Junos OS Evolved

Statement	Description
traceoptions	<p>The traceoptions option is deprecated at all hierarchy levels, except for the [edit protocols] hierarchy level.</p> <p>Starting in Junos OS Evolved Release 18.3R1, trace data from all applications on all nodes is collected on the Routing Engine in /var/log/traces. You can view collected traces with the show trace command. You can remove inactive tracing sessions with the clear trace command.</p> <p>NOTE: For Junos OS Evolved Release 18.3R1, the tracing feature is not supported for the [edit protocols] hierarchy level. For routing protocols, use the traceoptions option.</p>

CLI Commands Introduced in Junos OS Evolved

The changes in infrastructure between Junos OS and Junos OS Evolved sometimes require different CLI commands. For CLI commands introduced in Junos OS Evolved, see [Table 4 on page 12](#).

Table 4: CLI Commands Introduced in Junos OS Evolved

Command	Description	Link
clear security host-vpn security-associations	Clear host IPsec security association information. You can configure host IPsec with the [edit security host-vpn] statement.	<i>clear security host-vpn security-associations</i>

Table 4: CLI Commands Introduced in Junos OS Evolved (continued)

Command	Description	Link
clear trace	Clear traces on. The trace data from all nodes is collected on the Routing Engine in /var/log/traces .	<i>clear trace</i>
request system application	Start a specific application on the node you specify.	<i>request system application</i>
request system debug-info	Collect debug information from Junos OS Evolved, such as logs. The logs are stored in the /var/tmp/debug_collector_timestamp directory. A node option is provided, allowing for information from a specific node to be collected.	<i>request system debug-info</i>
request system shutdown	Halt, power off, or reboot the entire system.	<i>request system shutdown (halt power-off reboot)</i>
show security host-vpn security-associations	Display host IPsec security association information for a specific security association or for all connections. You can configure host IPsec with the host-vpn statement at the [edit security] hierarchy level.	<i>show security host-vpn security-associations</i>
show security host-vpn version	Display the version of IPsec being used in the system.	<i>show security host-vpn version</i>
show system applications	Display information about active applications on the system.	<i>show system applications</i>
show system errors	Display information about faults in the system. You can display all errors or the errors for one system component. Use this command to understand about faults and their correlation with other events.	<i>show system errors</i>
show system errors history	Display information about faults in the system that have been cleared. You can display all errors or the errors for one system component. Use this command to understand about faults and their correlation with other events.	<i>show system errors history</i>
show system node-attributes	Display information about the attributes of the current set of active and spare nodes on the system.	<i>show system node-attributes</i>
show system nodes	Display information about active nodes on the system.	<i>show system nodes</i>
show system software list	Display the installed versions on the Routing Engines in the system.	<i>show system software list</i>
show trace	Show the trace data from all nodes that is collected on the Routing Engine in /var/log/traces . You can refine how the trace data is shown by specifying trace time elapsed, application, process ID, and node. When disk usage of /var/log/traces exceeds the 5.7-GB limit, older trace files of applications are deleted to make space for new trace messages.	<i>show trace</i>

CLI Commands Modified in Junos OS Evolved

Some CLI commands in Junos OS Evolved have a different set of options than that in Junos OS. Sometimes there are little changes in behavior. See [Table 5 on page 14](#). For the CLI commands that produce changed output, see [Table 6 on page 16](#).

Table 5: CLI Commands Modified in Junos OS Evolved

Command	Description of Change in Junos OS Evolved	Link
<code>clear ipv6 neighbors</code>	In Junos OS Evolved, issuing the <code>clear ipv6 neighbors</code> command clears the cache for IPv6 neighbors in a reachable state.	<i><code>clear ipv6 neighbors</code></i>
<code>configure</code>	The <code>dynamic</code> option of the <code>configure</code> command is deprecated. The <code>configure dynamic</code> command is used to configure routing policies and certain routing policy objects in a dynamic database at the <code>[edit dynamic]</code> hierarchy level, a level you enter only by using the <code>configure dynamic</code> command. Because the <code>configure dynamic</code> command is deprecated, you cannot configure objects in a dynamic database, and you cannot use the <code>dynamic-db</code> statement.	<i><code>configure</code></i>
<code>ping</code>	The following options of the <code>ping</code> command are deprecated: <ul style="list-style-type: none"> • <code>detail</code> • <code>logical-system</code> • <code>loose-source</code> • <code>mac-address</code> • <code>strict</code> • <code>strict-source</code> • <code>vpls</code> 	<i><code>ping</code></i>
<code>request system software add</code>	The following options of the <code>request system software add</code> command are deprecated: <ul style="list-style-type: none"> • <code>best-effort-load</code> • <code>delay-restart</code> • <code>no-copy</code> • <code>on-primary</code> • <code>(re0 re1)</code> • <code>set</code> • <code>unlink</code> • <code>validate</code> • <code>validate-on-host</code> • <code>validate-on-routing-engine</code> 	<i><code>request system software add</code></i>

Table 5: CLI Commands Modified in Junos OS Evolved (continued)

Command	Description of Change in Junos OS Evolved	Link
<code>request system software rollback</code>	The following options are added to the <code>request system software rollback</code> command: <ul style="list-style-type: none"> • <code>(no-validate validate)</code> • <code>with-old-snapshot-config</code> 	<i>request system software rollback</i>
<code>request system storage cleanup</code>	A new option, <code>force-deep</code> , is added that cleans up all user-generated files as well.	<i>request system storage cleanup</i>
<code>show firewall</code>	The <code>application lsp</code> option is introduced, which you use to display implicit policers that are published by rpd.	<i>show firewall</i>
<code>show host</code>	The <code>routing-instance mgmt_junos</code> option is introduced.	<i>show host</i>
<code>show system connections</code>	The following options of the <code>show system connections</code> command are deprecated: <code>extensive</code> and <code>show-routing-instance</code> . The <code>node</code> option is introduced.	<i>show system connections</i>
<code>show system core-dumps</code>	The <code>node</code> option is introduced. the core dump files generated on the nodes are stored in the <code>/var/core/</code> directory.	<i>show system core-dumps</i>
<code>telnet</code>	The following options of the <code>telnet</code> command are deprecated: <ul style="list-style-type: none"> • <code>bypass-routing</code> • <code>interface</code> • <code>logical-system</code> • <code>no-resolve</code> • <code>source</code> 	<i>telnet</i>
<code>traceroute</code>	The following options of the <code>traceroute</code> command are deprecated: <ul style="list-style-type: none"> • <code>logical-system</code> • <code>next-hop</code> • <code>port</code> • <code>propagate-ttl</code> 	<i>traceroute</i>

Changed Command Output

For changes in output for Junos OS Evolved, see [Table 6 on page 16](#).

Table 6: Changed Command Output

Command	Description of Changed in Output	Link
clear interfaces statistics	Not only clears LACP statistics but also the counters displayed in the show lacp statistics interfaces command.	–
ping	When pinging a nonexistent route, the display output of the ping command does not print the number of packets sent or received or the number of packets loss.	<i>ping</i>
request system software delete	Output displays the version instead of the package.	<i>request system software delete</i>
request system software rollback	Output displays the version instead of the package.	<i>request system software rollback</i>
show chassis environment fpc	Displays different output.	<i>show chassis environment fpc</i>
show chassis fabric topology	Output displays UP for the State field, which should be OK .	–
show chassis fpc	The output displays only non-empty slots and the following modified slot states: Unknown , Onlining , Offlining , Standby , Fault , Fault-off , and Spare .	<i>show chassis fpc</i>
show interfaces	LACP packets on the members of an AE interface are not counted as part of the Bundle Input Statistics in the show interfaces ae number extensive command output.	<i>show interfaces (Aggregated Ethernet)</i>
show chassis pic status	The output displays only non-empty slots and the following modified slot states: Unknown , Onlining , Offlining , Standby , Fault , Fault-off , and Spare .	<i>show chassis pic</i>
show interfaces detail	Output displays the Last Flapped field with the value Never after a Routing Engine reboot. The Last Flapped field provides details of the date, time, and how long ago the interface went up. The value Never signifies that the interface never flapped.	<i>show interfaces detail</i>
show policer	Output displays only ARP policers. Use the show application lsp command to display implicit policers published by rpd.	<i>show policer</i> <i>show firewall</i>
show system statistics arp	After running ping on an unreachable host, output shows that counts for ARP requests received and for datagrams for an address no on the interface are incremented.	–
show system statistics tcp	Output for the show system statistics tcp command is trimmed to show only fields supported in Junos OS Evolved.	<i>show system statistics tcp</i>
show version	Output of the show version command is changed to clearly show which Junos architecture is running on the device.	<i>show version</i>

Table 6: Changed Command Output (continued)

Command	Description of Changed in Output	Link
	Output of the show version node all command is revised to explicitly identify the Routing Engine in both the XML and CLI output.	

Where to Find Information on Common Procedures

This guide, *Introducing Junos OS Evolved*, has information about the features and changes in the next generation of Junos OS. However, much about using Junos OS remains the same. Junos OS Evolved has the same CLI user interface, some of the same processes, and the same management and automation tools as classic Junos OS. You configure and manage Junos OS Evolved the same way as you always have configured and managed Junos OS.

For your convenience, this section lists some links to the Junos OS documentation you might want to consult.

- *QFX5200 Switch Hardware Guide*—Set up the QFX5200 switch.
- *Initial Router or Switch Configuration Using Junos OS*—Overview of initial configuration.
- *CLI User Guide*—Procedures on configuring and managing the CLI.
- *Getting Started Guide*—More procedures for initial configuration.
- *User Access and Authentication Feature Guide*—Procedures on granting access and setting up authentication on your device.
- *Network Management and Monitoring Guide*—Procedures on SNMP, remote monitoring (RMON), destination class usage (DCU) and source class usage (SCU) data, accounting profiles, and logging.
- *Installing Software Packages on QFX Series Devices*—Upgrade the release for a device already running Junos OS Evolved.

CHAPTER 2

Running Junos OS Evolved on a Device

- [Booting Junos OS Evolved by Using a Bootable USB Drive on page 19](#)

Booting Junos OS Evolved by Using a Bootable USB Drive

Junos OS Evolved supports booting from the USB device, that is, using a USB drive to boot Junos OS Evolved on a QFX5200-32C chassis. Booting from the USB device reformats the disk and reinstalls the software without prompting you. After the installation is done, the device waits for the USB drive to be removed from the USB port and then reboots into the new version.

There are several ways to create the Junos OS Evolved image on the USB drive. Also included are a procedure for booting from the USB drive and one for how to recover if the boot from the USB goes bad.

- [Create a Bootable USB Drive Using a Windows Machine on page 19](#)
- [Create a Bootable USB Drive Using a MAC OS X on page 21](#)
- [Create a Bootable USB Drive Using a Switch or Router Running Junos OS Evolved on page 22](#)
- [Boot Junos OS Evolved on the QFX5200 Switch on page 23](#)
- [Recover Junos OS Evolved Using USB Scratch Install on page 24](#)

Create a Bootable USB Drive Using a Windows Machine

You need the following items to perform this procedure:

- Windows desktop or laptop with a USB port.
- Version 2.0 or version 3.0 USB device with the following features:
 - USB device is big enough to hold the ISO image.
 - USB device must have no security features, such as a keyed boot partition.
- Junos OS Evolved ISO image

For a virtual Windows desktop you must map a physical USB of the host to the guest virtual machine (VM).

To create a bootable USB drive using a Windows machine:

1. Insert the USB drive into the Windows machine and format it using quick format.
2. Copy the ISO image to your laptop.
3. Extract the ISO Junos OS Evolved image using any extraction tools such as 7zip, WinZip, or WinRar.
4. Go to the directory you extracted the ISO to and copy the extracted ISO.
Select All (Ctrl+s) and Copy (Ctrl+c).
5. Paste the ISO to the USB drive (Ctrl+v).

7. Download **syslinux-6.04-pre1.zip**.

Syslinux helps you to make a USB bootable disk.

- a. Click this link:

<https://www.kernel.org/pub/linux/utils/boot/syslinux/Testing/6.04/syslinux-6.04-pre1.zip>

The syslinux-6.04-pre1.zip file is now in your **C:\Users\username\Downloads** directory.

- b. Extract syslinux-6.04-pre1.zip.

Right-click **syslinux-6.04-pre1.zip** and select **7-Zip > Extract to "syslinux-6.04-pre1"**.

These steps will extract the syslinux executable file to the **syslinux-6.04-pre1** directory under **Downloads**.

8. To create a master boot record (MBR) on the USB drive:
 - a. Type **cmd** in Windows search and right-click **Command Prompt** and select **Run as an Administrator**.
 - b. Using the Windows command prompt window, go to the extracted syslinux path, for example, **cd C:\Users\username\Downloads\syslinux-6.04-pre1**.
 - c. Run the following command, assuming your USB is in D drive:

```
syslinux64.exe --mbr --active --directory / --install D:
```

9. To set the label of the USB:
 - a. At the same command prompt, enter **label D:**, assuming that is the drive of your USB.

For example:

C:\Users\yourusername\Downloads\syslinux-6.04-pre1\bios\win64> **label D:**
You will see the following output:

```
Volume in drive D: is MISC
Volume Serial Number is 84F7-9F04
Volume label (11 characters, ENTER for none)?
```

b. After the ? prompt, enter **JUNOS** for the volume label.

10. Safely remove the USB drive and use it as a bootable USB drive on the device on which you plan to run Junos OS Evolved.

Create a Bootable USB Drive Using a MAC OS X

You need the following items to perform this procedure:

- A MAC OS X desktop or laptop with a USB port.
- Version 2.0 or version 3.0 USB device with following features:
 - USB device is big enough to hold the ISO image.

To create a bootable USB using MAC OS X:

1. Copy the install media (.img format) to the **/var/tmp/** directory of the routing device using the **scp** command.

For example:

```
$ scp user@server:/var/tmp/image-name /var/tmp/
password:
```

2. To get the list of devices on the MAC OS X device, run the **diskutil list** command.
3. Insert the USB flash drive into the USB port of the MAC OS X.
4. Run the **diskutil list** command again to determine the device node assigned to USB flash-drive (for example, **/dev/disk3**).
5. Run the **diskutil unmountDisk /dev/diskN** command.

Replace **N** with the disk number from the last command. (In this example, **N** would be 3.)

For example:

```
$ diskutil unmountDisk /dev/disk3
```

```
Unmount of all volumes on disk3 was successful
```

- Execute the command **sudo dd**
if=/var/tmp/junos-install-media-usb-srxhe-x86-64-17.4R1-S1.9.img of=/dev/rdiskN
bs=1m

For example:

```
$ sudo dd if=/var/tmp/usb.img of=/dev/rdisk3 bs=1m
```

```
Password:
```

```
965+0 records in
```

```
965+0 records out
```

```
1011875840 bytes transferred in 82.891882 secs (12207177 bytes/sec)
```

- The USB with image is created and ready for installation. Safely remove the USB drive and use it as a bootable USB drive on the device on which you plan to run Junos OS Evolved.

Create a Bootable USB Drive Using a Switch or Router Running Junos OS Evolved

You need the following items to perform this procedure:

- A switch or router with a USB port that is running Junos OS Evolved.
- Version 2.0 or version 3.0 USB device with following features:
 - USB device is big enough to hold the ISO image.
 - USB device must have no security features, such as a keyed boot partition.
 - USB device label should be **JUNOS**.

To create a bootable USB using a switch or router running Junos OS Evolved:

- Download **.img** image from Downloads site and copy it to the **/var/tmp/** directory of the switch or router running Junos OS Evolved using the **scp** command.

- Enter the shell as root:

```
user@switch> start shell user root
```

```
Password:
```

- Before inserting the USB device, list the contents of **/dev/**.

```
root@re0:~#ls /dev/sd*
```

```
/dev/sda /dev/sda3 /dev/sda6 /dev/sdb1 /dev/sdb4 /dev/sdb7
```

```
/dev/sda1 /dev/sda4 /dev/sda7 /dev/sdb2 /dev/sdb5
```

```
/dev/sda2 /dev/sda5 /dev/sdb /dev/sdb3 /dev/sdb6
```

```
root@re0:~#
```

- Insert the USB drive in the USB port.

- Repeat the command to list the contents of **/dev/**.

```
root@re0:~#ls /dev/sd*
/dev/sda   /dev/sda3  /dev/sda6  /dev/sdb1  /dev/sdb4  /dev/sdb7
/dev/sda1  /dev/sda4  /dev/sda7  /dev/sdb2  /dev/sdb5  /dev/sdc
/dev/sda2  /dev/sda5  /dev/sdb   /dev/sdb3  /dev/sdb6  /dev/sdc1
root@re0:~#
```



NOTE: **/dev/sdc** is the USB drive.

- Execute the following command, where **\$USB** identifies the device for that USB (typically **sdc** in Linux):

```
dd if=/var/tmp/usb.img of=/dev/$USB bs=100000
```

- The USB with image is created and ready for installation. Safely remove the USB drive and use it as a bootable USB drive on the device on which you plan to run Junos OS Evolved.

Boot Junos OS Evolved on the QFX5200 Switch

To perform this procedure, you must first create a USB drive with the Junos OS Evolved software image installed on it. For instructions, see [“Create a Bootable USB Drive Using a Windows Machine”](#) on page 19, [“Create a Bootable USB Drive Using a MAC OS X”](#) on page 21 or [“Create a Bootable USB Drive Using a Switch or Router Running Junos OS Evolved”](#) on page 22.

To install Junos OS Evolved on the QFX5200 switch using a USB drive:

- Connect to the console.
- Insert the USB drive with the Junos OS Evolved package in the **USB0** port on the routing device.
- Reboot the routing device from the CLI:

```
user@host> request system shutdown reboot usb
```

- To access the BIOS boot manager, press **Esc** while the system reboots.
The boot manager has the following two options:
 - USB: <model-name>
 - EFI USB Device
- From the boot manager, choose **USB: model-name**.

This choice ensures the boot uses the legacy BIOS boot mode, which is the option officially supported for Junos OS Evolved Release 18.3R1. After the device reboots, the software is copied onto the solid-state drive (SSD).

6. Remove the USB device.

Junos OS Evolved automatically installs.

Recover Junos OS Evolved Using USB Scratch Install

Problem **Description:** If, while you are trying to boot Junos OS Evolved from a USB device, the device goes to a bad state, follow this procedure.

Solution To recover using a USB scratch install:

1. Insert the bootable USB device into the device.
2. Go to the BIOS boot manager and select **USB: *model-name***.
 - a. Reboot the routing device.

```
user@host> request system shutdown reboot usb
```

- b. To access the BIOS boot manager, press ESC while the system reboots.

The scratch installation starts automatically and the operating system is installed.

3. Remove the USB or type a reboot command.