Juniper Networks | Engineering Simplicity

Introducing Junos OS Evolved

Published
2020-06-29

junos

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

Introducing Junos OS Evolved
20.2R1

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

1

CHAPTER

Overview of Junos OS Evolved

Junos OS Evolved Overview

Junos OS Evolved is next generation Junos OS. It is used just like Junos OS—the same CLI user interface, the same applications and features, the same management and automation tools—but its infrastructure is entirely modernized which enables higher availability, accelerated deployment, greater innovation and improved operational efficiencies.

Benefits

The development of Junos OS Evolved provides several benefits to Juniper Networks customers:

- Nearly all of the CLI and user interfaces are identical to those provided in Junos OS. This means there is virtually no learning curve in using Junos OS Evolved.
- It runs natively on Linux, providing direct access to all the Linux utilities and operations.
- All statistics and states are modeled and all states can be uniformly accessed. There is a central database which are used by not only Junos native applications but also external applications (using APIs).
- It has a fully distributed general-purpose software infrastructure that leverages all the compute resources on the network element, for example, CPUs in the Routing Engines, CPUs in the line cards, and potentially other x86 CPUs attached.

Native Linux Base

Whereas Junos OS runs over an instance of the FreeBSD operating system on a specific hardware element (for example, the CPU on the Routing Engine), Junos OS Evolved runs over a native Linux system. Having Linux as a base leverages a much wider, dynamic, and active development community. The Linux system also contains multiple third-party applications and tools developed for Linux that Junos OS Evolved can integrate with minimal effort.

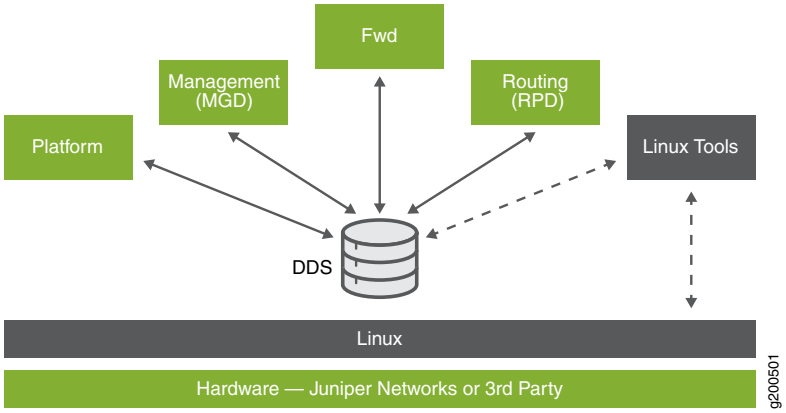
The Junos OS Evolved infrastructure is a horizontal software layer that decouples the application processes from the hardware on which they run. Effectively, this creates a general-purpose software infrastructure spanning all the different compute resources on the system (Routing Engine CPUs, line card CPUs, and possibly others). Application processes (protocols, services, and so on) run on top of this infrastructure and communicate with each other by publishing and consuming (that is, subscribing to) state.

Central Database for State

State is the retained information or status about physical or logical entities that is preserved, shared across the system, and supplied during restarts. State includes both operational and configuration state, including committed configuration, and interface state, routes, and hardware state. In Junos OS Evolved, all state is held in a central database called the Distributed Data Store (DDS).

The DDS does not interpret state. Its only job is to hold state received from subscribers and propagate state to consumers. It implements the publish-subscribe messaging pattern for communicating state between applications that are originators of a state to applications that are consumers of that state (see Figure 1 on page 7). Each application publishes state to and subscribes to state from the DDS directly, making applications independent of each other.

Figure 1: Publish-Subscribe Model



Decoupling applications in this manner isolates the failure of one application from others. The failing application can restart using the last known state of the system held in the state database.

Modular Design

Junos OS Evolved is composed of components with well-defined interfaces. This modular design means you replace at a component level, resulting in flexible packaging and faster upgrades. You can upgrade the system using an application-level restart, which, depending on the changes, in many instances will not require a system reboot. The system determines which modules have changed, and only changed modules are upgraded and restarted. Restarted applications reload the state that is preserved in the DDS.

Distributed Infrastructure

In Junos OS Evolved, a node is an entity composed of a physical compute resource such as a Routing Engine CPU or Line Card CPU or other x86 compute resource. Each node runs base Linux OS, state distribution pub-sub infrastructure, and applications that use this infrastructure.

The Junos OS Evolved infrastructure brings together multiple nodes into a single resource pool (that is. a distributed OS) wherein a Junos application can run on any node it is assigned to run on. This capability enables optimal resource utilization within an Junos Evolved system, and opens the door to new paradigms. Traditionally, the Routing Engines have x86 CPUs where Junos OS runs, and line cards, like FPCs or MPCs, have a general-purpose processor where the Junos Microkernel runs for Packet Forwarding Engine related functions. But in newer generations of x86 based line cards, a complete Linux operating system (OS) instance can be executed. Such line cards offer a more flexible framework and are capable of running processes and applications.

How Junos OS Evolved Differs from Junos OS

IN THIS SECTION

In many ways, Junos OS Evolved is the same as Junos OS: Key applications such as the routing, bridging, and management software is the same in both. And management plane interfaces and APIs, such as CLI, NETCONF, JET, JTI, AFI, and underlying data models, remain highly consistent. There are, however, some differences in behavior, the CLI syntax, and CLI and XML output. These differences are indicated throughout the Junos OS documentation. However, this section outlines the differences in one place, for your convenience. If applicable, a link takes you to the place in the Junos OS documentation that covers the item.

Behavioral Differences Between Junos OS Evolved and Junos OS

Behavioral differences between Junos OS Evolved and Junos OS are ways that the two operating systems act differently in certain circumstances. See Table 1 on page 9.

Table 1: How Junos OS Evolved Behavior Differs from Junos OS

Junos OS Evolved Behavior	Junos OS Behavior	Link to Documentation
Access and Authorization		
You must set up the password-less login between two devices to use **jcs:open** to open a connection to the local or remote device.	You are not limited to password-less login. Junos OS supports both a supplied password and interactive password, for example, to execute RPCs on remote device.	*open() Function (SLAX and XSLT)*
Interfaces		
Multiple releases of the software can be installed on the device simultaneously as long as there is space. If there is no more space, you must delete an older image of the software before installing the new one.	Only two versions of the software can be installed on the device: the current version and the previous version.	*Installing the Software Package on a Router with a Single Routing Engine*
The management interface name format changed to re0:mgmt-0/re0:mgmt-1. Both the management interfaces are configurable and displayed.	The management interface name that you use depends on the type of device that you are setting up. Some devices use me0, some use fxp0, and some use em0.	*Understanding Management Ethernet Interfaces*
When you issue the **show firewall filter ?** command, the names of the firewall filters are listed. The names of the Flowspec filters are not listed. To see the names of the configured Flowspec filters, use the **show firewall application routing** command.	When you issue the **show firewall filter ?** command, you see not only the names of the firewall filters listed but also the names of the configured Flowspec filters. The Flowspec filters show up inside underscores.	*show firewall*

Table 1: How Junos OS Evolved Behavior Differs from Junos OS *(continued)*

Junos OS Evolved Behavior	Junos OS Behavior	Link to Documentation
In an untagged LAG, child IFL members are created. Requests are made per child IFL member. The results are aggregated and displayed in the CLI. In a VLAN-tagged LAG, extra child IFLs are not created as part of the aggregated Ethernet bundle. Link IFL statistics and marker statistics for child IFLs are not displayed.	Child IFL members are created in untagged and VLAN-tagged LAGs. Requests are made per child IFL member. The results are aggregated and displayed in the CLI.	*Configuring Aggregated Ethernet Interfaces on PTX Series Packet Transport Routers*
When a new interface is added as a member to an AE bundle, the new member interface flaps: the physical interface is deleted as a regular interface and then added back in as an AE member and the statistics are reset.	When a new interface is added as a member to an AE bundle, that new interface is not first deleted as a lone interface and then added, but everything below it is. Because the interface is not deleted, it keeps all the statistics and other history associated with it.	*Configuration Guidelines for Aggregated Ethernet Interfaces* and *Understanding Aggregated Ethernet Interfaces and LACP for Switches*
For Junos OS Evolved, the software does not impose a limit on the maximum number of member (or child) interfaces in an aggregated interface.	For Junos OS, there is a limit of 64 member (or child) interfaces in an aggregated interface.	*Configuration Guidelines for Aggregated Ethernet Interfaces* and *Understanding Aggregated Ethernet Interfaces and LACP for Switches*
The command **show chassis fan** displays RPM in % measurement to indicate RPM speed.	The command **show chassis fan** displays RPM speed through indicated normal/high speed output.	*show chassis fan*
Messaging		
The process eventd does not give any warning message if there are duplicate policies. Instead eventd accepts the policy on a first-come, first-served basis.	The process eventd gives a warning message if you try to create duplicate policies.	*How Event Policies Work*
When the regular expression is to return empty matches, no error is message is displayed.	When the regular expression is to return empty matches, you get the following error: regex error: empty (sub)expression	*Junos System Log Regular Expression Operators for the match Statement*

Table 1: How Junos OS Evolved Behavior Differs from Junos OS *(continued)*

Junos OS Evolved Behavior	Junos OS Behavior	Link to Documentation
For op scripts run with the **max-datasize** configuration statement configured for the minimum, an error occurs. In Junos OS Evolved, the error is "Out of memory."	For op scripts run with the **max-datasize** configuration statement configured for the minimum, an error occurs. In Junos OS, the error is "Memory allocation failed."	*max-datasize*
Troubleshooting		
For Junos OS Evolved, a core file created during early bootup is stored in **/var/core/re**. But a core later in the bootup, for example, after the Routing Engine slot number can be determined, is stored in **/var/core/re0** or **/var/core/re1**. The command **show system core-dumps** continues to show all cores generated.	For Junos OS, cores files are stored in **/var/crash** or **/var/tmp**.	*show system core-dumps*
The **request system snapshot** command takes a snapshot of the contents of the **/soft** directory only.	The **request system snapshot** command takes a snapshot of the contents of the **/var/log**,**/var/core**, **/var/tmp**, and **/soft**directories.	*request system snapshot*, *Back Up the Currently Running and Active File System*, *Understanding How to Back Up an Installation on Switches*
The hierarchy **set system** scripts commit **traceoptions** does not exist. **traceoptions** is disabled for op, event, and commit scripts.	Use **traceoptions** to define tracing operations that track all routing protocol functionality in the routing device.	*traceoptions*
User Interface		

The menu used for root password recovery is the Grub Menu.
 	
`*Primary ptx-fixed-19.1-16`
`Primary [Recover password]`
`Primary-Rollback ptx-fixed-19.1-15`
 	
`Primary-Rollback [Recover password]`
	The menu used for root password recovery in Junos OS is the Junos Main Menu (the Recovery mode option).	*Recovering Root Password*

Table 1: How Junos OS Evolved Behavior Differs from Junos OS *(continued)*

Junos OS Evolved Behavior	Junos OS Behavior	Link to Documentation
The **show system firmware** command displays information based on the accessibility of the device, not the FRU state. The firmware information is cached so, even if the FRU is in a fault condition, the status from the **show system firmware** command appears as **OK**. But the fault is visible with the commands **show chassis alarms**, **show chassis fpc**, and so on.	When the FRU is offline, the cached firmware information of the FRU is not available to see.	*show system firmware*

New CLI Statements and Commands (Junos OS Evolved)

The changes in infrastructure between Junos OS and Junos OS Evolved sometimes require different CLI configuration statements and operational commands. For example, there is a new hierarchy level of statements in Junos OS Evolved that are not in Junos OS: **[edit security host-vpn]**. For more on these new statements and commands, see Table 2 on page 12.

Table 2: New CLI Statements and Commands (Junos OS Evolved)

Statement or Command	Description	Link
New Statements		
[edit security host-vpn]	Support for host IPsec in the control plane only (that is, IPsec between the router and external management devices, which is not available in Junos OS. This statement configures 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.	*Overview of IPsec* and *host-vpn*
[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: • **[edit security host-vpn connections *parent-connection-name* ike-proposal]** • **[edit security host-vpn connections *parent-connection-name* children *child-connection-name* esp-proposal]**	*connections (Host VPN)* and *children*

Table 2: New CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Description	Link
[edit security host-vpn connections children *child-name*]	Statements at this hierarchy level include **local-traffic-selector**, **remote**, and **remote-traffic-selector**.	*children*
[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.	*connections (Host VPN)*
[edit security host-vpn ike-log]	Statements at the **[edit security host-vpn]** hierarchy level used to configure IKE and IPsec values.	*ike-log*
[edit security host-vpn ike-secrets]	Statements at the **[edit security host-vpn]** hierarchy level used to configure IKE and IPsec values.	*ike-secrets*
[edit security host-vpn remote]	Configure identity details for authenticating the remote device during IKE negotiations.	*remote (Host VPN)*
[edit system trace application]	For Junos OS Evolved, trace data from all applications on all nodes is collected on the Routing Engine. You can view collected traces with the **show trace** command. You can remove inactive tracing sessions with the **clear trace** command.	*trace*
New Commands		
clear security host-vpn security-associations	Clear host IPsec security association information. You can configure host IPsec with the **[edit security host-vpn]** statement.	*clear security host-vpn security-associations*
clear trace	Junos OS Evolved uses a new tracing infrastructure. This command deletes the trace data stored on the Routing Engine.	*clear trace*
request system application	Start a specific process (for example cmdd) on the node you specify.	*request system application*
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. Use the **node** option to collect information from a specific node.	*request system debug-info*
request system software validate-restart	The command performs a dry run of the **request system software add restart** command and displays the ISSU impact of the new restart option. See *request system software add* for more on the **restart** option.	*request system software validate-restart*

Table 2: New CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Description	Link
restart	The following message will be logged when the **restart** command is used: **App restarting <app name>. Related apps that may be impacted - <related-app name>** .	*restart*
show chassis routing-engine hard-disk-test	Display the health of the hard disk. Use **disk** ***/dev/disk-name*** **status** to display the status of a particular disk.	*show chassis routing-engine*
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.	*show security host-vpn security-associations*
show security host-vpn version	Display the version of IPsec being used in the system.	*show security host-vpn version*
show system applications	Display information about active applications on the system.	*show system applications*
show system errors	Display information about faults in the system. **NOTE:** For Junos OS Evolved, only the QFX5200 supports this command. For all other Junos OS Evolved platforms, use the *show system errors active*, *show system errors count*, *show system errors error-id*, or *show system errors fru* command.	*show system errors*
show system errors history	Display information about faults in the system that have been cleared. **NOTE:** For Junos OS Evolved, only the QFX5200 supports this command. For all other Junos OS Evolved platforms, use the *show system errors active*, *show system errors count*, *show system errors error-id*, or *show system errors fru* command.	*show system errors history*
show system software add-restart	Display all console messages from the last in-service software upgrade (ISSU).	*show system software*

Table 2: New CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Description	Link
show system software list	Display the installed versions on the Routing Engines in the system.	*show system software list*
show system ztp	Junos OS Evolved implements ZTP using the Linux dhcp client. Users can find out the interfaces chosen by ZTP, arguments returned by DHCP, and ZTP state machine states.	*show system ztp*
show trace	Junos OS Evolved uses a new tracing infrastructure. This command shows the trace data from all nodes that are collected on the Routing Engine .	*show trace*

Modified CLI Statements and Commands (Junos OS Evolved)

Some CLI statements and commands in Junos OS Evolved have a different set of options from Junos OS. For a list of these changes, see Table 3 on page 15.

> **NOTE:** For the CLI commands that produce changed output, see Table 4 on page 20.

Table 3: Modified CLI Statements and Commands (Junos OS Evolved)

Statement or Command	Change in Junos OS Evolved	Link
Modified Statements		
[**edit interfaces *interface-name* ether-options**]	The following options are added to the **ether-options** statement: • fec • loopback-remote	*ether-options*
[**edit interfaces *interface-name* unit *logical-unit-number* vlan-id**]	The option **vlan-id 0** is not supported for this statement.	*vlan-id (VLAN ID to Be Bound to a Logical Interface)*
[edit system login password]	The **format** option for this statement is limited to the following options: **(sha256 | sha512)**.	*password*
Modified Commands		

Table 3: Modified CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Change in Junos OS Evolved	Link
clear ipv6 neighbors	In Junos OS Evolved, issuing the **clear ipv6 neighbors** command clears the cache for IPv6 neighbors in a reachable state.	*clear ipv6 neighbors*
configure	The **dynamic** option of the **configure** command is deprecated. The **configure dynamic** command is used to configure routing policies and certain routing policy objects in a dynamic database at the **[edit dynamic]** hierarchy level, a level you enter only by using the **configure dynamic** command. Because the **configure dynamic** command is deprecated, you cannot configure objects in a dynamic database, and you cannot use the **dynamic-db** statement.	*configure*
ping	The following options of the **ping** command are deprecated: • **detail** • **logical-system** • **loose-source** • **mac-address** • **strict** • **strict-source** • **vpls**	*ping*
request chassis routing-engine master switch	Starting in Junos OS Evolved Release 20.1R1, the default wait time on the PTX10008 between Routing Engine switchovers when using the **request chassis routing-engine master switch** command has increased from 120 seconds to 360 seconds.	*request chassis routing-engine master*

Table 3: Modified CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Change in Junos OS Evolved	Link
request system software add	The following options of the **request system software add** command are deprecated: • **best-effort-load** • **both-routing-engines** • **chassis** • **device-alias** • **delay-restart** • **force-host** • **lcc** • **member** • **no-copy** • **on-primary** • (**re0** | **re1**) • **re-choice** • **satellite** • **scc** • **set** • **sfc** • **upgrade-group** • **unlink** • **validate** • **validate_choice** • **validate-on-host** • **validate-on-routing-engine**	*request system software add*

Table 3: Modified CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Change in Junos OS Evolved	Link
request system software delete	The following options of the **request system software delete** command are deprecated: • **chassis** • **lcc** • **member** • **re-choice** • **scc** • **sfc** • **upgrade-group** • **unlink** • **validate** • **validate_choice** • **validate-on-host** • **validate-on-routing-engine**	*request system software delete*
request system software rollback	The following options are added to the **request system software rollback** command: • **(no-validate | validate)** • **with-old-snapshot-config** The following options are deprecated from the **request system software rollback** command: • **device-alias** • **satellite** • **satellite-arg** • **upgrade-group**	*request system software rollback*
request system software validate	The following options of the **request system software validate** command are deprecated: • **chassis** • **lcc** • **member** • **package-options** • **scc** • **sfc**	*request system software validate*

Table 3: Modified CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Change in Junos OS Evolved	Link
request system storage cleanup	A new option, **force-deep**, is added that cleans up all user-generated files as well.	*request system storage cleanup*
request system storage cleanup	The user is prompted to check the list of files to be deleted using the **dry-run** option. The following options are deprecated: • re0 • re1 • routing-engine	*request system storage cleanup*
set chassis error minor action	Starting in Junos OS Evolved Release 19.1R1, the **offline** and **disable-pfe** actions are not available for errors with minor severity.	*error*
show firewall	The **application lsp** option is introduced, which you use to display implicit policers that are published by rpd.	*show firewall*
show host	The **routing-instance mgmt_junos** option is introduced.	*show host*
show system connections	The following options of the **show system connections** command are deprecated: **extensive** and **show-routing-instance**. The **node** option is introduced.	*show system connections*
show system core-dumps	The **node** option is introduced. the core dump files generated on the nodes are stored in the **/var/core/** directory.	*show system core-dumps*
show system processes	The following options of the **show system processes** command are deprecated: • **esc-node** • **health** • **resource-limits**	*show system processes*

Table 3: Modified CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Change in Junos OS Evolved	Link
telnet	The following options of the **telnet** command are deprecated: • **bypass-routing** • **interface** • **logical-system** • **no-resolve** • **source**	*telnet*
traceroute	The following options of the **traceroute** command are deprecated: • **logical-system** • **next-hop** • **port** • **propogate-ttl**	*traceroute*

Changed CLI Command Output (Junos OS Evolved)

For changes in output for Junos OS Evolved, see Table 4 on page 20.

Table 4: Changed Command Output (Junos OS Evolved)

Command	Description of Change in Output	Link
clear interfaces statistics	Clears not only LACP statistics but also the counters displayed in the **show lacp statistics interfaces** command.	–
ping	When pinging a nonresponsive route, the display output of the **ping** command does not print the number of packets sent or received or the number of packets loss.	*ping*
request system snapshot	Output displays the names of the directory and the individual files being copied instead of only the directory names.	*request system snapshot*
show system snapshot	Output displays the snapshot device and a list of snapshots. The list shows the names of the snapshots instead of the version of the operating system. Output does not display the date the snapshot was created.	*show system snapshot*

Table 4: Changed Command Output (Junos OS Evolved) *(continued)*

Command	Description of Change in Output	Link
request system software delete	Output displays the version instead of the package.	*request system software delete*
request system software rollback	Output displays the version instead of the package.	*request system software rollback*
The **show chassis environment cb** command does not show the Bus and FPGA revision information. Use the **show system firmware** command in order to view the FPGA revision or version information for the CB.	Use the **show chassis environment cb** command to display environmental information about the Control Boards (CBs).	*show chassis environment cb*
show chassis environment fpc	Displays different output.	*show chassis environment fpc*
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.	*show interfaces (Aggregated Ethernet)*
show interfaces	Configuration of IPv6 over the re0:mgmt-* interfaces is supported.	-
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.	*show interfaces detail*
show interfaces extensive	Output does not display the Packet Forwarding Engine configuration and CoS default bandwidth allocation information.	*show interfaces (M Series, MX Series, T Series Routers, and PTX Series Management and Internal Ethernet)*
show multicast route extensive	Output displays the **Sensor ID** field that corresponds to a multicast route.	*show multicast route*
show multicast usage	Output displays the **Sensor ID** field that corresponds to a multicast route.	*show multicast usage*

Table 4: Changed Command Output (Junos OS Evolved) *(continued)*

Command	Description of Change in Output	Link
show snmp mib walk	The **show snmp mib walk jnxFilledDescr** output only shows the fan tray number. This output does not show the number of fan slots present in each tray.	*show snmp mib*
show snmp mib get	Output for a Routing Engine displays the Routing Engine slot number, not the Routing Engine number.	*show snmp mib*
show system errors fru detail	Output displays status of FRUs including CB, chassis, fans, FPC, FPM, PDU, PICS, PSM, RE, and SIB, not just FPC.	*show system errors fru*
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.	*show system statistics tcp*
show system uptime	Output displays only the **System booted** and **System-wide users** information. The output does not display information on current time, system booted, protocols started, or last configured parameters. The **show system uptime node** command shows the other information	*show system uptime*
show task replication	Output displays the same state whether the command is run from the master or spare Routing Engine.	*show task replication*
show version	Output of the **show version** command is changed to clearly show which Junos architecture is running on the device. Output of the **show version node all** command is revised to explicitly identify the Routing Engine in both the XML and CLI output.	*show version*
traceroute	Output of the **traceroute** command displays MPLS data parsed in the same way as the Linux **traceroute** command: L=label, E=exp_use, S=stack_bottom, and T=TTL.	*traceroute*

Removed CLI Statements and Commands (Junos OS Evolved)

For a listing of which CLI statements and commands are removed from Junos OS Evolved, see Table 5 on page 23. Where there is an alternative statement or command to use, it is noted in the table.

Table 5: Removed CLI Statements and Commands (Junos OS Evolved)

Statement or Command	Description
Removed Statements	
gigether-options	Starting in Junos OS Evolved Release 20.1R1, the **gigether-options** statement at the **[edit interfaces *interface-name*]** hierarchy no longer appears because it is not needed. To configure link aggregation groups (LAG), use the **set interfaces *interface-name* ether-options** command instead.
edit system services extension-service notification	Notification service for JET applications is not supported in Junos OS Evolved.
traceoptions	The **traceoptions** option is removed from many of the hierarchy levels. Routing protocols (the **[edit protocols]** hierarchy level) is one of the applications still using **traceoptions**.
Removed Commands	
request system software abort	Deprecated. There is no alternate command replacing it. The **request system software add** command has a built-in feature not to start an upgrade if a reboot is pending after an upgrade or rollback.
request system software (add | delete) set	Deprecated. Because for Junos OS Evolved all packages are bundled into one single ISO file, the **set** option serves no purpose in the **request system software add** and **request system software delete** commands.
request system software in-service-upgrade	Deprecated. Use the **request system software add restart** command for ISSU. The **request system software add** command has a built-in feature not to start upgrade if a reboot is pending after an upgrade or rollback.
request system software set	Deprecated. To set the current system to an installed software version, use the **request system software rollback reboot** command.
request system storage user-disk	Deprecated. There are no satellite packages in Junos OS Evolved.
set date	Setting the date/time manually is not supported. NTP is the only supported method to maintain/set time. If you issue the **set date** command, an error occurs.

Table 5: Removed CLI Statements and Commands (Junos OS Evolved) *(continued)*

Statement or Command	Description
show chassis fabric unreachability	Deprecated. See the **show system errors** command for similar functionality.
show chassis fabric summary	The **show chassis fabric summary** command is removed. See the **show system errors** command for similar functionality.
show chassis network-services	Deprecated.
show chassis routing-engine errors	This command has been replaced by **show system errors** in Junos OS Evolved.
show class-of-service forwarding-table	Deprecated. The removed options include **classifier**, **classifier mapping**, **drop-profile**, **policer**, **rewrite-rule**, **rewrite-rule mapping**, **scheduler-map**, and **shaper**.
show database-replication	Deprecated.
show interfaces em0 | em1	The em0 and em1 Ethernet management interfaces are removed. Use re0:mgmt-* for Routing Engine 0 (Routing Engine 1 would be re1:mgmt-*).
show interfaces ixgbe0 | ixgbe1	The ixgbe0 and ixgbe1 internal interfaces are removed.
show interfaces mac-database	Deprecated. The **MAC accounting and policing not supported** message is displayed.
show pfe	Deprecated.
show system software detail	Deprecated. Use **show system software list** to display a list of the software versions installed on all nodes. For more details about the software, use **show version detail**.
show system switchover	Deprecated.
set system services xnm-clear-text	Command is not supported and has been deprecated from Junos OS Evolved.

XML Differences Between Junos OS and Junos OS Evolved

This section lists the differences in XML output between Junos OS and Junos OS Evolved.

system storage cleanup

In Junos OS, the output of this command uses the **file** XML tag for all file types in the list of files to be deleted. In Junos OS Evolved, the output of this command groups different file types inside different XML tags.

system storage cleanup | display XML (Junos OS)

`user@host>` **request system storage cleanup | display xml**

```
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/18.4I0/junos">
    <system-storage-cleanup-information>
        <file-list junos:style="normal">
            <file>
                    <file-name>/var/log/dfcd_enc.0.gz</file-name>
                    <size junos:format="551B">551</size>
                    <date>Nov 23 15:33</date>
            <file>
        </file-list>
    </system-storage-cleanup-information>
</rpc-reply>
```

system storage cleanup | display XML (Junos OS Evolved)

`user@host>` **request system storage cleanup | display xml**

```
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/19.1I0/junos">
    <system-storage-cleanup-information>
        <node>
            <node-name> RE0 </node-name>
            <core-file-list>
                <description>List of all core files to be cleared: </description>

                <file>

<file-name>/var/core/re0/auditd.re.re0.17130.2019_02_28.03_39_36.tar.gz</file-name>

                    <size>3.8M</size>
                    <date>Thu Feb 28 03:40</date>
                </file>
            </core-file-list>
            <core-local-host-file-list>
            </core-local-host-file-list>
            <core-subdir-file-list>
```

```
            </core-subdir-file-list>
            <fpc-file-list>
            </fpc-file-list>
            <logical-systems-file-list>
            </logical-systems-file-list>
            <log-file-list>
                <description>Clears all App logs, App traces and App SI traces
under /var/log/*, /var/log/traces/* and /var/log/si_traces/* </description>
            </log-file-list>
            <iso-file-list>
            </iso-file-list>
        </node
    </system-storage-cleanup-information>
</rpc-reply>
```

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 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.

- *Initial Router or Switch Configuration Using Junos OS*—Overview of initial configuration.
- *Getting Started Guide for Junos OS*—More procedures for initial configuration.
- *User Access and Authentication User 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 the Software Package on a Router with a Single Routing Engine*—Procedure on installing Junos OS on a device with one Routing Engine.
- *Junos OS Installation Packages Prefixes*—Overview of install packages by prefix, including Junos OS Evolved images.

2

CHAPTER

Booting Junos OS Evolved from a USB

Booting Junos OS Evolved by Using a Bootable USB Drive

IN THIS SECTION

You can boot Junos OS Evolved from a USB device. 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 Device

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 device:

1. Install Win32 Disk Imager on your laptop or computer.

   You can download it from https://sourceforge.net/projects/win32diskimager/.

2. Download the required Junos OS image from the Downloads page to the Documents directory of your laptop or computer.

3. Insert a USB flash drive into the USB port of your laptop or computer.

4. Open the win32diskimager application and, in the **Image File** box, type the path to the Documents directory (or click the folder icon to navigate to the Documents directory) and select the install media image.

5. Under **Device**, select the USB flash-drive and click **Write and Confirm**. The Progress box shows the progress.

6. Remove the USB flash drive once it is complete.

   The USB flash-drive is now ready to use as a bootable disk.

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 MAC OS 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/disk*N*** 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
   ```

6. Execute the command **sudo dd if=/var/tmp/junos-evo-install-ptx-fixed-x86-64-19.2R1.31-EVO.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)
   ```

7. 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:

1. 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.

2. Enter the shell as root:

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

   ```
   Password:
   ```

3. 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:~#
   ```

4. Insert the USB drive in the USB port.

5. 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.

6. 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
```

7. 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 from a Bootable USB Drive

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 MAC OS X" on page 29 or "Create a Bootable USB Drive Using a Switch or Router Running Junos OS Evolved" on page 30.

To install Junos OS Evolved on a device that runs Junos OS Evolved using a USB drive:

1. Connect to the console.

2. Insert the USB drive with the Junos OS Evolved package in the **USB0** port on the routing device.

3. Reboot the routing device from the CLI:

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

   When the reboot and installation of the Junos OS Evolved package is complete, you have a choice as to running a snapshot or not:

   ```
   Installation of image junos-evo-install-ptx-fixed-x86-64-19.2R1.31-EVO done.
   Boot version is now 'junos-evo-install-ptx-fixed-x86-64-19.2R1.31-EVO'
   Do you want to run snapshot on secondary ssd? (Y/N)
   ```

4. Do one of the following:

   - Enter **Y** to perform snapshot.

     Output is:

     ```
     Do you want to run snapshot on secondary ssd? (Y/N)Y
         performing snapshot...
         Mounting version junos-evo-install-qfx-ms-fixed-x86-64-19.2R1.31-EVO...
     ```

   - Enter **N** to skip taking a snapshot. The previous snapshot is kept.

Output is:

```
Do you want to run snapshot on secondary ssd? (Y/N)N
    Setting next_boot
    Booting from 0000
```

5. 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. Access the BIOS manager to check the USB selection:

   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.

3. In the BIOS boot manager, select one of the following:

   - For PTX10003 devices, select **EFI USB**.
   - For QFX5200 devices, select **USB: *model-name***.

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

4. Remove the USB or type a reboot command.

3

CHAPTER

Signing 3rd Party Applications with Junos OS Evolved

Protecting the Integrity of Junos OS Evolved with IMA

Network devices that run Junos OS Evolved are protected by an integrity solution called Integrity Measurement Architecture (IMA).

Integrity is a fundamental security property that represents trust, completeness, and freedom from alteration. In computer security, common targets for integrity protections are operating system files. A common method of ensuring integrity is to compare a file against a known good file.

In the context of Junos OS Evolved, the security goal is to ensure that the software running on a device has not been accidentally or maliciously altered. The software running on a device is either authentic Junos software from Juniper Networks or authorized software deployed by a customer.

The threat model for network devices includes attempts by malicious actors to deploy malware that violates either the implicit or explicit policies of device owners. Such malware could include back doors, Trojan horses, or implants that could adversely the safe and secure operation of devices or networks. Malicious actors use a variety of tools, techniques, and procedures to breach integrity including physical attacks, local attacks, and remote attacks.

Many regulatory schemes levy file integrity requirements, including PCI-DSS - Payment Card Industry Data Security Standard (Requirement 11.5), SOX - Sarbanes-Oxley Act (Section 404), NERC CIP - NERC CIP Standard (CIP-010-2), FISMA - Federal Information Security Management Act (NIST SP800-53 Rev3), HIPAA - Health Insurance Portability and Accountability Act of 1996 (NIST Publication 800-66) and the SANS Critical Security Controls (Control 3).

In order to ensure file integrity and to mitigate the malware risk, Junos OS Evolved runs IMA, and a companion mechanism: the Extended Verification Module (EVM). These open source protections are part of a set of Linux Security Modules that are industry-standard and consistent with the trust mechanisms specified by the Trusted Computing Group.

Juniper Networks applies digital signatures to Junos OS Evolved files, and allows customers to apply digital signatures as well. Digital signatures are created using protected private keys, and then verified using public keys embedded into one or more keyrings.

The IMA/EVM subsystem protects the system by performing run-time checks. If a file fails verification, it is not opened or executed.

That means that unverified software is blocked on a device running Junos OS Evolved.

In order to run customer applications, Juniper Networks offers the capability for authorized customer operators to sign and add software that IMA will verify.

RELATED DOCUMENTATION

Signing Third-Party Applications to Run Natively

IN THIS SECTION

Signing Keys Overview

Starting in Junos OS Evolved Release 20.1R1, you can generate signing keys and use them to sign executable files or shared objects. Signing an executable file gives it permission to run on the device, allowing you to approve trusted applications to run alongside authorized Juniper Networks software.

Junos OS Evolved requires users to sign all files that will be mapped into memory for execution. This includes the following file types:

- Executable and Linkable Format (ELF) files
- Shared Objects (.so) files

The following types of files do not need to be signed:

- Docker containers
- Applications inside containers
- Scripts

> **NOTE:** Although scripts don't need to be signed, they do need to be passed through a signed interpreter for execution. Junos OS Evolved comes installed with signed Python 2 and Python 3 interpreters that can be used through the **python *script-name*** shell command.

Signing keys are controlled by a Linux subsystem called Integrity Measurement Architecture (IMA). IMA policy consists of rules that define which actions needs to be taken before a file can be executed. IMA measurement policy will measure and store a file's hash, and IMA appraisal policy will make sure that the file has a valid hash or digital signature. IMA will only allow a file to run if this validation succeeds. For more information about IMA, see "Protecting the Integrity of Junos OS Evolved with IMA" on page 35.

Signing keys are stored in the *system keystore*, and the certificates used the verify signing keys are stored in the *IMA extended keyring*. Keep reading to learn how to generate, import, view, and use signing keys.

Generating Signing Keys

IN THIS SECTION

Keys can be generated through the OpenSSL command-line or a OpenSSL configuration file.

Generating Signing Keys Using the OpenSSL Command-Line

The following example OpenSSL command can be used to generate signing keys:

```
openssl req -new \
 -newkey rsa:3072 \    # Create an RSA 3072 key
 -x509 \               # Need an X509 certificates
 -sha256 \             # Strong hashing algorithm
 -nodes \              # No encrypted private-key
 -out ima-cert.x509 \  # Name of the certificate file
 -outform DER \        # Key in DER format
 -keyout privkey.pem \ # Name of the private key
```

This command will generate 2 files:

1. **privkey.pem** - The PEM encoded private key that can be used to sign executable files.
2. **ima-cert.x509** - The DER encoded certificate to be loaded into the IMA extended keyring.

> **NOTE:** The OpenSSL command-line is limited in its functionality. It does not allow you to set values for the X509v3 extensions. All keys generated using the command above can be used as Certificate Authorities (CAs), and therefore can be used to sign other certificates. To prevent this, we can use an OpenSSL Configuration File.

Generating Signing Keys Using an OpenSSL Configuration File

Create a file named **ima-x509.cnf** and paste the following contents:

```
# Begining of ima-x509.cnf
[ req ]
default_bits = 2048
distinguished_name = custom_distinguished_name
prompt = no
string_mask = utf8only
x509_extensions = custom_exts

[ custom_distinguished_name ]
O = Juniper Networks, Inc.
CN = IMA extended signing key
```

```
emailAddress = john.smith@juniper.net

[ custom_exts ]
basicConstraints=critical,CA:FALSE
keyUsage=digitalSignature
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid
# EOF
```

After the configuration file is created, use the following OpenSSL command to create the **ima-privkey.pem** and **ima-cert.x509** files:

```
openssl req -new \
        -nodes \
        -utf8 \
        -sha1 \
        -days 36500 \
        -batch \
        -x509 \
        -config ima-x509.cnf \
        -outform DER -out ima-cert.x509 \
        -keyout ima-privkey.pem
```

The private key file **ima-privkey.pem** is used to generate signing keys, and the certificate file **ima-cert.x509** is used to verify the signature. Both files are used during the process of importing signing keys into the system keystore and IMA extended keyring.

Importing Signing Keys into the System Keystore and IMA Extended Keyring

Signing keys need to be imported into the system keystore prior to use. Keys that are imported into the system keystore are automatically imported into the IMA extended keyring.

To import a signing key into the system keystore, use the **request security system-keystore import** command with the following 3 mandatory arguments:

1. **key-name** - A unique name for the key
2. **private-key** - Path to the private key file
3. **x509-cert** - Path to the DER encoded certificate file

The following example command will create a key named **ima-test-key** by using the private key file **ima-privkey.pem** and the certificate file **ima-cert.x509**:

```
user@host> request security system-keystore import key-name ima-test-key private-key
 ima-privkey.pem x509-cert ima-cert.x509

  Key Name:               ima-test-key
  Private Key Path:       /etc/ima-ext/ima-test-key/privkey.pem
  X509 Cert Path:         /etc/ima-ext/ima-test-key/ima-cert.x509
  Key SKI:                b71b35e380517cd224b46072dadeb6c53e0a58a1
```

When the key is successfully imported into the **system-keystore** you will see the above output displaying the name of the key, the paths to the private key and certificate on disk, and the Subject Key Identifier (SKI) for the key. You can check if this SKI matches with the key loaded into the IMA Extended keyring with the following command:

```
user@host> show security integrity extended-keyring

Keyring
 351716837 ---lswrv      0     0  keyring: ima_ext
 684930381 --als--v      0     0   \_ asymmetric: Juniper Extended Signing Key:
b71b35e380517cd224b46072dadeb6c53e0a58a1
```

Viewing the System Keystore and IMA Extended Keyring

You can view the contents of the system keystore and the IMA extended keyring through Junos OS Evolved CLI **show** commands.

Use the **show security integrity system-keystore** command to view the available signing keys in the system keystore:

```
user@host> show security integrity system-keystore

  Available signing keys:
  ---
  Key Name:               ima-test-key
  Private Key Path:       /etc/ima-ext/ima-test-key/privkey.pem
  X509 Cert Path:         /etc/ima-ext/ima-test-key/ima-cert.x509
  Key SKI:                b71b35e380517cd224b46072dadeb6c53e0a58a1
  ---
  Key Name:               test-key1
  Private Key Path:       /etc/ima-ext/test-key1/privkey.pem
  X509 Cert Path:         /etc/ima-ext/test-key1/ima-cert.x509
  Key SKI:                332f173d61bba03fed5399a609523cbd3cfe66b3
  ---
  Key Name:               test-key2
  Private Key Path:       /etc/ima-ext/test-key2/privkey.pem
  X509 Cert Path:         /etc/ima-ext/test-key2/ima-cert.x509
  Key SKI:                26ebafd58b54f7b8b530d0311503fd84873ee754
  ---
```

The information in the Key SKI field can be used to map these keys to the IMA extended keyring.

Use the **show security integrity extended-keyring** command to view the contents of the IMA extended keyring:

```
user@host> show security integrity extended-keyring

Keyring
 351716837 ---lswrv      0     0  keyring: ima_ext
 684930381 --als--v      0     0   \_ asymmetric: Juniper Extended Signing Key:
b71b35e380517cd224b46072dadeb6c53e0a58a1
 316767440 --als--v      0     0   \_ asymmetric: Juniper Extended Signing Key:
26ebafd58b54f7b8b530d0311503fd84873ee754
 950431262 --als--v      0     0   \_ asymmetric: Juniper Extended Signing Key:
332f173d61bba03fed5399a609523cbd3cfe66b3
```

How to Sign Applications

After a signing key has been imported into the system keystore, it can be used to sign executable binaries.

Use the **request security integrity measure file *filename* key *key-name*** command to sign a file.

The following example command shows a file named **ima-test** being signed by a key named **ima-test-key**:

```
user@host> request security integrity measure file ima-test key ima-test-key
Successfully signed file /data/var/home/root/ima-test
```

You can verify that your file was successfully signed by using the **request security integrity appraise file *filename* key *key-name*** command, as follows:

```
user@host> request security integrity appraise file ima-test key ima-test-key

File /data/var/home/root/ima-test has a valid IMA signature
```

If the file was not signed properly, the following message will display:

```
user@host> request security integrity appraise file ima-test key ima-test-key
 warning: IMA signature verification failed for /data/var/home/root/ima-test using
ima-test-key
 IMA appraisal for /data/var/home/root/ima-test failed.
```

After a file has been signed, it can be run natively on your Junos OS Evolved device.

How to Run Signed Applications

On attempting to execute a file that has not been signed, you may get a **Permission Denied** error:

```
user@host:~# ./ima-test
-sh: ./ima-test: Permission denied
```

Once the file has been successfully signed, it can then be executed from a shell prompt by adding the **./** prefix in front of the filename:

```
user@host:~# ./ima-test
Hello, World!
```