Release Notes: Junos® PyEZ Release 2.1.2

Release 2.1.2
29 August 2017

Junos PyEZ can be used with the following Juniper Networks® hardware: ACX Series, EX Series, M Series, MX Series, NFX Series, PTX Series, QFabric systems, QFX Series, SRX Series, and T Series.

These release notes accompany Juniper Networks Junos PyEZ Release 2.1.2. They describe new and changed features, limitations, and known and resolved problems in the software.

DISCLAIMER

Use of the Junos PyEZ software implies acceptance of the terms of this disclaimer, in addition to any other licenses and terms required by Juniper Networks.

Juniper Networks is willing to make the Junos PyEZ software available to you only on the condition that you accept all of the terms contained in this disclaimer. Please read the terms and conditions of this disclaimer carefully.

The Junos PyEZ software is provided *as is*. Juniper Networks makes no warranties of any kind whatsoever with respect to this software. All express or implied conditions, representations and warranties, including any warranty of non-infringement or warranty of merchantability or fitness for a particular purpose, are hereby disclaimed and excluded to the extent allowed by applicable law.

In no event will Juniper Networks be liable for any direct or indirect damages, including but not limited to lost revenue, profit or data, or for direct, special, indirect, consequential, incidental or punitive damages however caused and regardless of the theory of liability arising out of the use of or inability to use the software, even if Juniper Networks has been advised of the possibility of such damages.

Contents

Copyright © 2017, Juniper Networks, Inc.

Copyright © 2017, Juniper Networks, Inc.

New and Changed Features

This section describes the new features and enhancements to existing features in Junos PyEZ Releases 2.0.1, 2.1.0, 2.1.1, and 2.1.2.

Config Class

- **Support for load update configuration operations**—Starting in Junos PyEZ Release 2.1.0, you can load configuration data on a device running Junos OS using a **load update** operation by including the **update=True** argument in the **Config load()** method. This operation is equivalent to the Junos OS CLI **load update** configuration mode command, which compares a complete loaded configuration against the existing configuration and replaces only those portions of the configuration that have changed. During the commit operation, only system processes that are affected by changed configuration elements parse the new configuration.

  [See Using Junos PyEZ to Configure Devices Running Junos OS.]

Device Class

- **New Device properties**—Starting in Junos PyEZ Release 2.1.0, the **Device** class supports the **master** and **re_name** properties. The **master** property returns **True** if the Routing Engine to which the application is connected is the master Routing Engine, and **False** otherwise. The **re_name** property returns a string specifying the name of the Routing Engine to which the application is connected.

  [See Understanding Junos PyEZ Device Facts and Connection Properties.]

- **Device.connected property**—Starting in Junos PyEZ Release 2.1.0, the **Device.connected** attribute is converted to a property so that it returns the current state of the connection rather than the static value obtained for the initial connection. A **SessionListener** monitors the session and responds to transport errors by raising a TransportError exception and setting the **Device.connected** property to **False**.

  [See Understanding Junos PyEZ Device Facts and Connection Properties.]

Copyright © 2017, Juniper Networks, Inc.

- **New console_has_banner parameter**—Starting in Junos PyEZ Release 2.1.0, you can include **console_has_banner=True** in the **Device** argument list to telnet to a console server that emits a banner message and then requires the user to press **Enter** after the message to retrieve the login prompt. When you include the **console_has_banner=True** argument and the application does not receive a login prompt upon initial connection, the application waits for 5 seconds and then emits a newline (**\n**) character so that the console server issues the login prompt. If you omit the argument and the connection hangs, the application instead emits the **<close-session/>** RPC to terminate the connection.

  [See Connecting to Devices Running Junos OS Using Junos PyEZ.]

Exception Handling

- **New exceptions**—Starting in Junos PyEZ Release 2.1.0, the following new exception is supported:
  - JSONLoadError—Exception raised when a device returns malformed JSON in an RPC reply.
- **Support for suppressing RpcError exceptions raised in response to warnings**—Starting in Junos PyEZ Release 2.1.0, you can include the **ignore_warning** argument in a method call or RPC invocation to suppress **RpcError** exceptions that would be raised in response to warnings. The **ignore_warning** argument can take the Boolean value **True** to ignore all warnings, or it can take a string or list of strings specifying which warnings to ignore.

  [See Suppressing RpcError Exceptions Raised for Warnings in Junos PyEZ Applications.]

FS Class

- **Support for retrieving disk usage information**—Starting in Junos PyEZ Release 2.1.0, the **jnpr.junos.utils.fs.FS** class provides the **directory_usage** method, which returns disk usage information for files and directories on the device.

  [See directory_usage().]

Management

- **Support for on-demand fact gathering**—Starting in Junos PyEZ Release 2.1.0, device facts are gathered on demand for all connection types. Each fact is gathered and cached the first time you access its value or the value of a dependent fact. To refresh the facts, call the **Device facts_refresh()** method. #638

  [See Understanding Junos PyEZ Device Facts and Connection Properties.]
- **Support for fact gathering for Junos Device Manager and NFX250 Network Services Platforms**—Starting in Junos PyEZ Release 2.1.0, Junos PyEZ can gather facts from Junos Device Manager (JDM) and from NFX250 Network Services Platforms.

Copyright © 2017, Juniper Networks, Inc.

SCP Class

- **Support for private SSH key file**—Starting in Junos PyEZ Release 2.0.1, when you include the **ssh_private_key_file** parameter in the **Device** argument list to define a specific SSH private key file for authentication, the **SCP** instance also uses the same key file for authentication when transferring files.

  [See Transferring Files Using Junos PyEZ.]

StartShell Class

- **New timeout parameter**—Starting in Junos PyEZ Release 2.0.1, you can include the **timeout** parameter in the **StartShell** argument list to specify the duration of time in seconds that the utility must wait for the expected string or pattern in the Junos OS shell before timing out. If you do not specify a timeout, the default is 30 seconds.

  [See Accessing the Unix Shell Using Junos PyEZ.]

- **Support for executing nonreturning shell commands**—Starting in Junos PyEZ Release 2.1.0, the **StartShell run** method supports setting the **this** argument to the value **None** to enable execution of nonreturning commands in the shell. When you include the **this=None** argument, the method waits until the specified timeout value before retrieving and returning all output on the shell. If you omit the argument or set it equal to a specific string or pattern, the method might return partial output for a nonreturning command if it encounters a default prompt or the specified string pattern within the command output.

  [See Accessing the Unix Shell Using Junos PyEZ.]

SW Class

- **Support for rebooting all Routing Engines**—Starting in Junos PyEZ Release 2.1.0, the **jnpr.junos.utils.sw.SW reboot()** takes the **all_re** parameter to specify which Routing Engines to reboot. To reboot all Routing Engines in a dual Routing Engine setup or a Virtual Chassis setup, include **all_re=True** or omit the parameter. To reboot only the Routing Engine to which the application connects, include **all_re=False**. In earlier releases, by default, the **reboot()** method only reboots the Routing Engine to which the application connects.

  [See Rebooting a Device Running Junos OS Using Junos PyEZ.]

- **ISSU and NSSU support**—Starting in Junos PyEZ Release 2.1.0, you can include the **issu=True** or **nssu=True** argument in the **sw.install()** method to perform a unified in-service software upgrade (unified ISSU) or a nonstop software upgrade (NSSU), respectively, on devices that support and also meet the requirements for the feature.

  [See Using Junos PyEZ to Install Software on Devices Running Junos OS.]

Copyright © 2017, Juniper Networks, Inc.

Remote Procedure Calls

- **Support for retrieving configuration data for standard and custom YANG data models**—Starting in Junos PyEZ Release 2.1.0, the **get_config()** RPC supports the **model** and **namespace** parameters for retrieving configuration data corresponding to standard and custom YANG data models that have been added to a device running Junos OS. To retrieve configuration data corresponding to custom, OpenConfig, or IETF YANG data models, set the **get_config() model** argument to **'custom'**, **'ietf'**, or **'openconfig'**, respectively.

  [See Using Junos PyEZ to Retrieve the Configuration.]

Tables and Views

- **New bgpTable and Module Tables**—Starting in Junos PyEZ Release 2.1.0, the Junos PyEZ package includes two new operational Table modules. The **bgp** module includes the **bgpTable** Table, which executes the **get-bgp-neighbor-information** RPC to retrieve BGP neighbor information. The **inventory** module includes the **Module** Table, which executes the **get-chassis-inventory** RPC to retrieve chassis hardware information.

  [See Junos PyEZ Predefined Operational Tables and Views.]

- **New Tables in the ospf module**—Starting in Junos PyEZ Release 2.1.0, the **ospf** operational Table module includes two new Tables. The **ospfTable** Table executes the **get-ospf-overview-information** RPC to retrieve OSPF summary information, and the **OspfRoutesTable** Table executes the **get-ospf-route-information** RPC to retrieve information about OSPF routes.

  [See Junos PyEZ Predefined Operational Tables and Views.]

- **New fields for LLDPNeighborView**—Starting in Junos PyEZ Release 2.1.0, the **lldp** operational Table module, which includes **LLDPNeighborTable** and **LLDPNeighborView**, includes two new fields in **LLDPNeighborView**. The **remote_port_id** and **remote_sysname** fields map to the **lldp-remote-port-id** and **lldp-remote-system-name** elements, respectively, in the RPC output.

Installation

This section contains information about installing Junos PyEZ on the configuration management server.

Copyright © 2017, Juniper Networks, Inc.

- **Junos PyEZ Dockerfile**—Juniper Networks provides a Junos PyEZ Dockerfile, which automatically builds a Junos PyEZ Docker image for every Junos PyEZ release. The Docker container is a lightweight, self-contained system that bundles Junos PyEZ, its dependencies, and Python into a single portable container. The Docker image enables you to quickly run Junos PyEZ in interactive mode, as an executable package, or as a terminal on any platform that supports Docker.

  

  NOTE: **The latest Junos PyEZ Docker image is built using the most recently committed code in the Junos PyEZ source repository, which is under active development and might not be stable.**

  [See Installing Junos PyEZ and Examples of using the Docker image.]

Changes in Behavior and Syntax

This section lists the changes in behavior of Junos PyEZ features and changes in Junos PyEZ syntax from Junos PyEZ Release 2.1.0.

- **Device.connected attribute converted to a property**—Starting in Junos PyEZ Release 2.1.0, the **Device.connected** attribute is converted to a property so that it returns the current state of the connection rather than the static value obtained for the initial connection.
- **Change for reboot() method default**—Starting in Junos PyEZ 2.1.0, the **reboot()** method reboots all Routing Engines in a dual Routing Engine or Virtual Chassis setup. In earlier releases, the **reboot()** method only reboots the Routing Engine to which the application connects.
- **Facts gathered on demand**—Starting in Junos PyEZ Release 2.1.0, device facts are gathered on demand for all connection types. Each fact is gathered the first time you access its value or the value of a dependent fact. In earlier releases, when you call the **Device open()** method to connect to a device, Junos PyEZ automatically gathers the device facts for NETCONF-over-SSH connections and gathers the device facts for Telnet and serial console connections when you explicitly include **gather_facts=True** in the **Device** argument list.

Known Behavior

This section contains the known behavior, system maximums, and limitations in software in Junos PyEZ Release 2.1.2.

Management

- **Software installation**—The software installation process provided by the **jnpr.junos.utils.sw** module is currently designed to support simple deployment scenarios. The expected use case for this software is deploying new equipment.

Copyright © 2017, Juniper Networks, Inc.

The Junos PyEZ software installation process supports the following scenarios:

- Standalone devices with a single Routing Engine
- Standalone devices equipped with dual Routing Engines
- EX Series Virtual Chassis in mixed and non-mixed-mode configurations
- QFX Series Virtual Chassis in mixed and non-mixed-mode configurations
- Mixed EX Series and QFX Series Virtual Chassis
- Deployment configurations that have some form of *in-service* features enabled, such as unified ISSU or NSSU

The Junos PyEZ software installation process has the following scenarios available as Beta:

- MX Series Virtual Chassis
- SRX Series chassis clusters
- Virtual Chassis Fabric (VCF)

Known Issues

For the current list of known issues in the Junos PyEZ software, see the open issues listing for the Junos PyEZ project in GitHub at https://github.com/Juniper/py-junos-eznc/issues.

Resolved Issues

This section lists the issues fixed in the Junos PyEZ releases. For the complete and most current list of resolved issues in the Junos PyEZ software, see the closed issues listing for the Junos PyEZ project in GitHub at https://github.com/Juniper/py-junos-eznc/issues?q=is:issue+is:closed.

- Resolved Issues: Release 2.1.2 on page 8
- Resolved Issues: Release 2.1.1 on page 9
- Resolved Issues: Release 2.1.0 on page 9
- Resolved Issues: Release 2.0.1 on page 9

Resolved Issues: Release 2.1.2

- Facts are not gathered correctly from SRX Series Services Gateways in chassis cluster configurations. #697, #698
- The **SW()** class does not correctly handle software upgrades on multichassis systems and devices with multiple Routing Engines because of the new-style fact gathering. #700

Copyright © 2017, Juniper Networks, Inc.

- Junos PyEZ does not clearly indicate when a device returns invalid JSON in an RPC reply. It should throw a proper exception. #701, #706
- Calls to **lock()** and **unlock()** can incorrectly raise a **ConnectClosedError** exception in some circumstances. #708

Resolved Issues: Release 2.1.1

- After the implementation of the **ignore_warning** argument, in some circumstances, Junos PyEZ does not return the same value that would have been returned prior to the introduction of **ignore_warning**. #691

Resolved Issues: Release 2.1.0

- Special characters in the output for console connections can result in an uncaught **lxml.etree.XMLSyntaxError** exception. #610
- In the predefined operational Tables **PhyPortTable** and **PhyPortDiagTable**, the default value for the **interface_name** argument does not include 100-Gigabit Ethernet interfaces, which use et- as the media type in the interface name. #609
- Fact gathering halts if there is an issue retrieving a model number or serial number. #615
- The **pprint** function does not print device facts in pretty print manner. #660, #661
- When warnings are suppressed by **ignore_warnings**, Junos PyEZ should return a normal RPC response. #688

Resolved Issues: Release 2.0.1

- The **Config** class **lock()** method returns an **AttributeError** exception when the **Device** argument list includes the **normalize=True** argument. #587, #590
- The **StartShell** class **run** function throws a **UnicodeDecodeError** exception for special characters. #588, #589
- RPCError exceptions are not handled correctly for console connections. #592, #595

Documentation and Release Notes

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

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Copyright © 2017, Juniper Networks, Inc.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/

Copyright © 2017, Juniper Networks, Inc.

- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

Revision History

29 August 2017—Revision 1, Junos PyEZ Release 2.1.2

Copyright © 2017 Juniper Networks, Inc. All rights reserved.

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. and/or its affiliates in the United States and other countries. All other trademarks may be 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.