

Release Notes: Junos[®] PyEZ Release 1.2.3

Release 1.2.3
7 October 2015

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

These release notes accompany Juniper Networks Junos PyEZ Release 1.2.3. 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 upon 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

New and Changed Features	3
Config Class	3
Device Class	3
Exception Handling	4
Management	4
Remote Procedure Calls	5
Tables and Views	6

- Changes in Behavior and Syntax 6
- Known Behavior 6
 - Management 7
- Known Issues 7
- Resolved Issues 7
 - Resolved Issues: Release 1.2.3 8
 - Resolved Issues: Release 1.2.2 8
 - Management 8
 - Remote Procedure Calls 8
 - Resolved Issues: Release 1.2.1 8
 - Resolved Issues: Release 1.2.0 8
 - Management 9
 - Remote Procedure Calls 9
 - Tables and Views 9
- Documentation and Release Notes 9
- Documentation Feedback 9
- Requesting Technical Support 10
 - Self-Help Online Tools and Resources 10
 - Opening a Case with JTAC 10
- Revision History 11

New and Changed Features

This section describes the new features and enhancements to existing features in Junos PyEZ Release 1.2.3.

- [Config Class](#)
- [Device Class](#)
- [Exception Handling](#)
- [Management](#)
- [Remote Procedure Calls](#)
- [Tables and Views](#)

Config Class

- **Commit options support**—Starting with Junos PyEZ Release 1.2, the `commit()` method in the `jnpr.junos.utils.config.Config` utility supports the following parameters. To use the parameter, set it equal to `True` in the `commit()` method argument list.
 - **detail**—Displays detailed messages about the commit operation after it completes.
 - **force_sync**—Copies the candidate configuration stored on the requesting Routing Engine to the other Routing Engine, verifies the candidate's syntactic correctness, and commits it on both Routing Engines even if there are open configuration sessions or uncommitted configuration changes on the other Routing Engine. Configuration sessions on the other Routing Engine are terminated before the configuration is synchronized and committed.
 - **sync**—Copies the candidate configuration stored on the local Routing Engine to the other Routing Engine, verifies the candidate's syntactic correctness, and commits it on both Routing Engines.

[See [Using Junos PyEZ to Commit the Candidate Configuration](#).]

Device Class

- **normalize parameter**—Starting with Junos PyEZ Release 1.2, you can include the `normalize=True` argument when you create a `Device` instance to normalize RPC replies for the duration of a session with a device. Normalizing the reply strips out all leading and trailing whitespace and replaces sequences of internal whitespace characters with a single space, which can make it easier to retrieve information using XPath expressions that select for a specific node or value.

[See Normalizing the XML RPC Reply in [Using Junos PyEZ to Execute RPCs on Devices Running Junos OS](#).]

- **ssh_config parameter**—Starting with Junos PyEZ Release 1.2, you can include the `ssh_config` parameter when you create a `Device` instance to direct Junos PyEZ to query a specific SSH configuration file. If you omit this parameter, Junos PyEZ queries the default SSH configuration file at `~/.ssh/config` if one exists.

[See [Using Junos PyEZ to Connect to and Retrieve Facts from Devices Running Junos OS](#).]

- **ProxyCommand support**—Starting with Junos PyEZ Release 1.2, you can use ProxyCommand to access the target device running Junos OS through an intermediary host that supports netcat. When you initiate a connection to the device, Junos PyEZ queries the SSH configuration file, which can include ProxyCommand configurations.
[See [Using Junos PyEZ to Connect to and Retrieve Facts from Devices Running Junos OS.](#)]
- **Serialization of Junos OS facts**— Starting with Junos PyEZ Release 1.2, custom encoders enable you to view a YAML or JSON representation of the entire set of facts for a device. By default, the `Device` instance returns the device facts as a Python dictionary.
[See [Using Junos PyEZ to Connect to and Retrieve Facts from Devices Running Junos OS.](#)]

Exception Handling

- **New exceptions**—Starting with Junos PyEZ Release 1.2, the following new exceptions are supported:
 - `ConfigLoadError`—Exception that gets raised when a load configuration operation fails and that provides information as to why the operation failed.
 - `ConnectClosedError`—Exception that gets raised when the connection to a device is unexpectedly terminated or when the `close()` method is called for a device instance that does not have an active connection with a device. When the exception is thrown, `device.connected` is set to `False`.
 - `RPCTimeoutError`—Exception that gets raised when an RPC times out and that provides information about the RPC, the device, and the timeout value.
- **Enhanced exceptions**—Starting with Junos PyEZ Release 1.2, the following existing exceptions have been enhanced:
 - `CommitError`—Exception now includes field descriptors and information as to why the commit failed.
 - `ConnectError`—Exception now includes any underlying error message in addition to the exception type and host information.
 - `RpcError`—Exception now includes field descriptors in the error message.

Management

- **Mixed Virtual Chassis upgrade support**—Starting with Junos PyEZ Release 1.2, the `install()` method in the `jnpr.junos.utils.sw.SW` utility supports the `pkg_set` argument, which enables you to supply a list of the installation packages required for upgrading members in a mixed Virtual Chassis.
[See [Using Junos PyEZ to Install Software on Devices Running Junos OS.](#)]
- **Support for force_host parameter (QFX5100 switch)**—Starting with Junos PyEZ Release 1.2, the `install()` method in the `jnpr.junos.utils.sw.SW` utility supports the `force_host` argument for QFX5100 switches. Including `force_host=True` forces the addition of the host software package or bundle (ignoring warnings) on the switch. The parameter is identical to the `force-host` option for the `request system software add` operational command.

[See [jnpr.junos.utils.sw.](#)]

- **SCP ProxyCommand support**—Starting with Junos PyEZ Release 1.2, the `jnpr.junos.utils.scp.SCP` utility supports ProxyCommand, which enables you to transfer files from the local host to the target device through an intermediary host that supports netcat.

[See [Transferring Files Using Junos PyEZ.](#)]

Remote Procedure Calls

- **Support for specifying the format of RPC output**—Starting with Junos PyEZ Release 1.2.3, you can specify the format of the RPC output returned by the `Device` instance `rpc` property as text or JSON instead of the default XML format. To specify the format, include the `{'format':'json'}` or `{'format':'text'}` dictionary as the first argument in the argument list. RPC output in text format is enclosed within an `<output>` element. JSON format is supported for devices running Junos OS Release 14.2 or a later release.

[See Executing RPCs as a Property of the Device Instance in [Using Junos PyEZ to Execute RPCs on Devices Running Junos OS.](#)]

- **Support for specifying the RPC timeout interval**—Starting with Junos PyEZ Release 1.2, you can specify the RPC timeout value when you execute individual RPCs using the `Device` instance `rpc` property. Increasing the timeout interval can help prevent the RPC from timing out during execution. Including the `dev_timeout=seconds` argument sets the device timeout value for that single operation only. By default, RPCs time out after 30 seconds.

[See Executing RPCs as a Property of the Device Instance in [Using Junos PyEZ to Execute RPCs on Devices Running Junos OS.](#)]

- **Support for retrieving Junos XML RPC tags**—Starting with Junos PyEZ Release 1.2, the `jnpr.junos.device.Device` class includes the `display_xml_rpc` method, which retrieves the Junos XML RPC tags for the given operational mode command. You can map the tags to the corresponding Junos PyEZ RPC method, and execute the RPC using the `Device` instance `rpc` property.

[See Mapping Junos OS Commands to Junos PyEZ RPCs in [Using Junos PyEZ to Execute RPCs on Devices Running Junos OS.](#)]

- **Support for XML normalization**—Starting with Junos PyEZ Release 1.2, you can normalize an RPC reply, which strips out all leading and trailing whitespace and replaces sequences of internal whitespace characters with a single space. Normalizing the RPC reply can make it easier to retrieve information using XPath expressions that select for a specific node or value. You can use `normalize=True` to enable normalization for the duration of a session with a device or to normalize an individual RPC reply when you execute the RPC. By default, normalization is not enabled.

[See Normalizing the XML RPC Reply in [Using Junos PyEZ to Execute RPCs on Devices Running Junos OS.](#)]

Tables and Views

- **Configuration Table and View support**—Starting with Junos PyEZ Release 1.2, you can define configuration Tables and Views to retrieve and analyze configuration data from a device running Junos OS. You can load configuration Tables from either the device or an XML file or object.

[See [Defining Junos PyEZ Configuration Tables](#).]

- **Op Table normalization**—Starting with Junos PyEZ Release 1.2, op Tables normalize Table values as well as keys. You can disable normalization by including **normalize: false** under the Table **args** property or by including **normalize=False** as an argument to the Table instance **get()** method.

[See [Defining Junos PyEZ Operational Tables](#) and [Using Junos PyEZ Operational Tables and Views](#).]

- **Regular expression support for View Boolean values**—Starting with Junos PyEZ Release 1.2, Views support setting field item values to a Boolean, which can be set based on regular expressions.

[See [Defining Junos PyEZ Views for Operational Tables](#) and [Defining Junos PyEZ Views for Configuration Tables](#).]

- **Table keys support for the XPath union operator**—Starting with Junos PyEZ Release 1.2, Table keys can use the XPath union (|) operator, which can be used to specify an implicit "or" and is useful in situations where different tag names are present for different types of configurations or releases.

[See [Defining Junos PyEZ Operational Tables](#).]

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 1.2.

- **Format optional for configuration strings**—As of Junos PyEZ Release 1.2, the `jnpr.junos.utils.config.Config` class `load()` method no longer requires the **format** attribute when you load configuration data as a string. Junos PyEZ automatically detects the format.
- **Composite Table keys**—As of Junos PyEZ Release 1.2, composite Table keys that use XPath expressions replace any missing elements with **None**. Prior to Junos PyEZ Release 1.2, missing elements would cause the program to throw an exception.

Known Behavior

This section contains the known behavior, system maximums, and limitations in software in Junos PyEZ Release 1.2.

- **Management**

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.

The following scenarios are supported:

- 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 do not have any form of "in-service" features enabled, such as unified ISSU and NSSU

The following scenarios are known *not* to be supported:

- MX Series Virtual Chassis
- SRX Series chassis clusters
- Virtual Chassis Fabric (VCF)
- Deployment configurations that have some form of "in-service" features enabled, such as unified ISSU or NSSU

Known Issues

For the most 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> .

- When you include the `detail=True` argument in the `jnpr.junos.utils.config.Config.commit()` method, Junos PyEZ might generate an exception for invalid XML.

Resolved Issues

This section lists the issues fixed in the Junos PyEZ main release and maintenance releases. For the 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 1.2.3 on page 8](#)
- [Resolved Issues: Release 1.2.2 on page 8](#)
- [Resolved Issues: Release 1.2.1 on page 8](#)
- [Resolved Issues: Release 1.2.0 on page 8](#)

Resolved Issues: Release 1.2.3

- The `jnpr.junos.device.Device` class incorrectly prioritizes the private keys specified using `IdentityFile` in the SSH config file over the keys explicitly defined using the `ssh_private_key_file` argument in the `Device` constructor.
- Facts gathering raises an exception for empty serial numbers on certain M Series routers, VMX devices, and node devices in a QFabric system.
- Facts gathering raises a `KeyError` exception for TX Matrix Plus Routing Matrix configurations.
- The `jnpr.junos.utils.sw.SW` class incorrectly identifies multiple Virtual Chassis members for some standalone devices that are capable of participating in but are not members of a Virtual Chassis.
- The `jnpr.junos.utils.sw.SW` class generates an error regarding an incorrect number of arguments for the `update_progress()` function even when the correct number of arguments is provided.
- The `jnpr.junos.utils.scp.SCP` class should use the same defaults for printing log messages as the `SW` module and should only print at given intervals.

Resolved Issues: Release 1.2.2

- [Management](#)
- [Remote Procedure Calls](#)

[Management](#)

- Paramiko logger warnings are generated in some situations.
- `Device` password setter fails to set password.
- Facts gathering fails for standalone EX8200 switches.

[Remote Procedure Calls](#)

- Converting RPC command output from XML to JSON using the `json.dumps()` method fails when the contents include an XML comment.

Resolved Issues: Release 1.2.1

- Unable to use `get()` method multiple times with Tables.

Resolved Issues: Release 1.2.0

- [Management](#)
- [Remote Procedure Calls](#)
- [Tables and Views](#)

Management

- Device facts do not account for all physical Routing Engines in an MX Series Virtual Chassis.
- The Junos PyEZ session times out when gathering facts on an MX Series Virtual Chassis.
- Device facts falsely indicate dual Routing Engines for standalone devices that are capable of participating in a Virtual Chassis.
- `jnpr.junos.utils.Utils.start_shell.StartShell` throws an exception when the user is root.

Remote Procedure Calls

- Junos PyEZ throws an exception when converting RPC command output from XML to JSON using the `json.dumps()` method.

Tables and Views

- Op Tables do not load properly when you load locally stored XML data from a file or `lxml` object.
- Junos PyEZ throws an exception when converting Views that have compound keys from XML to JSON using the `json.dumps()` method.

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>.

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/>
- 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

7 October 2015—Revision 1, Junos PyEZ Release 1.2.3

Copyright © 2016, Juniper Networks, Inc. All rights reserved.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, 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.