



# Juniper Extension Toolkit API Guide

Release

1.0



Modified: 2016-07-10

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

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.

*Juniper Extension Toolkit API Guide*

1.0

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

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

YEAR 2000 NOTICE

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

## **END USER LICENSE AGREEMENT**

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

# Table of Contents

	About the Documentation . . . . .	vii
	Documentation and Release Notes . . . . .	vii
	Supported Platforms . . . . .	vii
	Documentation Conventions . . . . .	vii
	Documentation Feedback . . . . .	ix
	Requesting Technical Support . . . . .	x
	Self-Help Online Tools and Resources . . . . .	x
	Opening a Case with JTAC . . . . .	x
<b>Chapter 1</b>	<b>Overview of the JET APIs . . . . .</b>	<b>13</b>
	JET API Overview . . . . .	13
<b>Chapter 2</b>	<b>JET Notification API . . . . .</b>	<b>15</b>
	JET Notification API Overview . . . . .	15
	JSON Format of JET Notification Messages . . . . .	15
	Subscribing to Events . . . . .	16
	Understanding the JET Notification API . . . . .	16
	Programming JET Notification for Non-Python Languages . . . . .	18
<b>Chapter 3</b>	<b>JET Service APIs . . . . .</b>	<b>21</b>
	JET Service APIs Overview . . . . .	21
	Interfaces Service API . . . . .	21
	Firewall Service API . . . . .	21



# List of Tables

	<b>About the Documentation</b> .....	<b>vii</b>
	Table 1: Notice Icons .....	viii
	Table 2: Text and Syntax Conventions .....	viii
<b>Chapter 2</b>	<b>JET Notification API</b> .....	<b>15</b>
	Table 3: Junos Event Topics .....	16
	Table 4: MQTT Libraries by Language .....	19



# About the Documentation

- Documentation and Release Notes on page vii
- Supported Platforms on page vii
- Documentation Conventions on page vii
- Documentation Feedback on page ix
- Requesting Technical Support on page x

## 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>.

## Supported Platforms

---

For the features described in this document, the following platforms are supported:

- MX104
- MX2010
- MX2020
- MX240
- MX480
- MX80
- MX960

## Documentation Conventions

---

Table 1 on page viii defines notice icons used in this guide.

Table 1: Notice Icons







Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page viii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  user@host> <b>configure</b>
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> <b>show chassis alarms</b>  No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> <li>Introduces or emphasizes important new terms.</li> <li>Identifies guide names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul>	<ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS CLI User Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name:  [edit] root@# <b>set system domain-name</b> <i>domain-name</i>



Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> <li>To configure a stub area, include the <b>stub</b> statement at the <b>[edit protocols ospf area area-id]</b> hierarchy level.</li> <li>The console port is labeled <b>CONSOLE</b>.</li> </ul>
< > (angle brackets)	Encloses optional keywords or variables.	<b>stub &lt;default-metric metric&gt;;</b>
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	<b>broadcast   multicast</b> <b>(string1   string2   string3)</b>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	<b>community name members [ community-ids ]</b>
Indentation and braces ( { } )	Identifies a level in the configuration hierarchy.	<b>[edit]</b> routing-options { static { route default { nexthop address; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
<b>GUI Conventions</b>		
<b>Bold text like this</b>	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> <li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li> <li>To cancel the configuration, click <b>Cancel</b>.</li> </ul>
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select <b>Protocols&gt;Ospf</b> .

## 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](mailto: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>.



## CHAPTER 1

# Overview of the JET APIs

- [JET API Overview on page 13](#)

## JET API Overview

---

The Juniper Extension Toolkit (JET) has the following APIs:

- Python JetHandler class

Establish connection with JET through the **JetHandler** interface.

- [JET Notification API Overview on page 15](#)

Provides interfaces in the JETHandler Python package that allow you to subscribe to events and designate a callback function to receive events when they occur.

- [JET Service APIs Overview on page 21](#)

Provides interfaces to access the control plane on the device and a management interface to run operational and configuration commands.

### Related Documentation

- [JET Notification API Overview on page 15](#)
- [JET Service APIs Overview on page 21](#)
- [JetHandler API](#)



## CHAPTER 2

# JET Notification API

- [JET Notification API Overview on page 15](#)

## JET Notification API Overview

---

The JET Notification API includes interfaces in the **JETHandler** Python package that allow you to subscribe to events and designate a callback function to receive events when they occur. These Python interfaces provide an interface to the MQTT notification system that for languages other than Python must be handled outside of JET (see “[Programming JET Notification for Non-Python Languages](#)” on page 18). For details about MQTT and Python, see <http://mosquitto.org/>.

- [JSON Format of JET Notification Messages on page 15](#)
- [Subscribing to Events on page 16](#)
- [Understanding the JET Notification API on page 16](#)
- [Programming JET Notification for Non-Python Languages on page 18](#)

## JSON Format of JET Notification Messages

JET notification is delivered in JSON format. The JSON message has two parts, one the header, and other the event-information portion, called *attributes*. The header is common for all events. It contains the event ID, hostname, time, severity, and facility of the event. The attributes vary depending event’s topic name.

All kernel rtsock events will have **info** as the severity and **KERNEL** as the facility. For syslog events, the severity and facility will be same as that of the Junos OS syslog messages.

The following is an example event JSON file.

```
“jet-event”: {
  “event-id”: “KERNEL_EVENT_IFD_ADD”
  “hostname”: “mydevice”,
  “time”: “2016-01-07”,
  “severity”: “info”,
  “facility”: “KERNEL”,
  “attributes”: {
    “name”: “ge-0/0/0”,
    “snmp-id”: 520,
    “flags”: 8
  }
}
```

}

## Subscribing to Events

Applications developed using JET can subscribe to the events listed in [Table 3 on page 16](#). A *topic* is an endpoint to which the clients connect. A topic acts as the central distribution hub for publishing and subscribing messages. Topics are simple, hierarchical strings, encoded in UTF-8, delimited by a forward slash.

**Table 3: Junos Event Topics**

Events	Topic	Event Information Returned
Physical Interface (IFD)	<ul style="list-style-type: none"> <li><code>/junos/events/kernel/interfaces/ifd/add//ifdname</code></li> <li><code>/junos/events/kernel/interfaces/ifd/change/ifdname</code></li> <li><code>/junos/events/kernel/interfaces/ifd/delete/ifdname</code></li> </ul>	name, snmp-id, flags
Logical Interface (IFL)	<ul style="list-style-type: none"> <li><code>/junos/events/kernel/interfaces/ifl/add/iflname</code></li> <li><code>/junos/events/kernel/interfaces/ifl/change/iflname</code></li> <li><code>/junos/events/kernel/interfaces/ifl/delete/iflname</code></li> </ul>	name, subunit, snmp-id, flags
Family (IFF)	<ul style="list-style-type: none"> <li><code>/junos/events/kernel/interfaces/iff/add/iflname/family-type</code></li> <li><code>/junos/events/kernel/interfaces/iff/change/iflname/family-type</code></li> <li><code>/junos/events/kernel/interfaces/iff/delete/iflname/family-type</code></li> </ul>	name, subunit, family, table-name, flags
Address	<ul style="list-style-type: none"> <li><code>/junos/events/kernel/interfaces/ifa/add/iflname/family-type/address</code></li> <li><code>/junos/events/kernel/interfaces/ifa/change/iflname/family-type/address</code></li> <li><code>/junos/events/kernel/interfaces/ifa/delete/iflname/family-type/address</code></li> </ul>	name, subunit, family, local-address, destination-address, broadcast-address, flags
Firewall	<ul style="list-style-type: none"> <li><code>/junos/events/kernel/firewall/filter/add/filtername</code></li> <li><code>/junos/events/kernel/firewall/filter/change/filtername</code></li> <li><code>/junos/events/kernel/firewall/filter/delete/filtername</code></li> </ul>	name, version, client-id, filter-type, protocol, interface-name, flags
Route	<ul style="list-style-type: none"> <li><code>/junos/events/kernel/route/add/family/prefix-with-length</code></li> <li><code>/junos/events/kernel/route/change/family/prefix-with-length</code></li> <li><code>/junos/events/kernel/route/delete/family/prefix-with-length</code></li> </ul>	table-name, logical-router-name, address-family, route-type, route-prefix, arrayof(next-hop-address), flags
Route-table	<ul style="list-style-type: none"> <li><code>/junos/events/kernel/route-table/add/tablename/lname</code></li> <li><code>/junos/events/kernel/route-table/change/tablename/lname</code></li> <li><code>/junos/events/kernel/route-table/delete/tablename/lname</code></li> </ul>	name, logical-router-name, address-family, flags
Syslog	<code>/junos/events/syslog/event-id</code>	arrayof(attribute-value pairs)

## Understanding the JET Notification API

The JET Notification APIs are as follows:

**OpenNotificationSession()**—Creates a request-response session with the JET server.

Raises an exception in the case of invalid arguments or when JET notification server is not accessible.



The user is required to configure allow-clients IP addresses using CLI. By default the behavior is to disallow all clients to connect to broker unless user has configured to allow them. The **bind\_address** argument passed in the **OpenNotificationSession()** call needs to be configured for the **allow-clients** configuration statement in device to authorize the connection at mosquito broker side.

#### Arguments

- **device**—JET Server IP address. Default is localhost.
- **port**—JET Notification port number. Default is 1883.
- **user**—Username on the JET server, used for authentication and authorization.
- **password**—Password to access the JET server, used for authentication and authorization.
- **keepalive**—Maximum period in seconds between communications with the broker.
- **bind\_address**—Client address to bind. Can be used to control access at broker side.

**CloseNotificationSession()**—Closes the notification connection with the broker. After this method is executed successfully, further calls over this session will fail.

**Subscribe()**—Subscribes to a specific topic the client application is interested in.

**Unsubscribe()**—Takes the topic as argument and unsubscribes to the given topic. If topic name is not given as argument (no arguments), this method unsubscribes to all the topics that the application is already subscribed for.

**CreateIFDTopic()**—Creates physical interface topic.

#### Arguments

- **op**—Operation (**ADD, DELETE, CHANGE, ALL**). Default is **ALL**.
- **ifd\_name**—The physical interface name to subscribe to. Default is all physical interface names.

**CreateIFLTopic()**—Creates logical interface topic.

#### Arguments

- **op**—Operation (**ADD, DELETE, CHANGE, ALL**). Default is **ALL**.
- **ifd\_name**—The physical interface name to subscribe to. Default is all physical interface names.
- **sub\_unit**—This argument is mandatory when an interface name is passed.
- **family\_type**—Default is all family types.

**CreateIFFTopic()**—Creates family topic.

#### Arguments

- **op**—Operation (**ADD, DELETE, CHANGE, ALL**). Default is **ALL**.

- **ifd\_name**—The physical interface name to subscribe to. Default is all physical interface names.

**CreateIFATopic()**—Creates address topic.

**Arguments**

- **op**—Operation (**ADD, DELETE, CHANGE, ALL**). Default is **ALL**.
- **ifd\_name**—The physical interface name to subscribe to. Default is all physical interface names.
- **sub\_unit**—This argument is mandatory when an interface name is passed.
- **family\_type**—Default is all family types.
- **address**—Default is all addresses.

**CreateFirewallTopic()**—Creates firewall topic.

**Arguments**

- **op**—Operation (**ADD, DELETE, CHANGE, ALL**). Default is **ALL**.
- **filter\_name**—Filter name to subscribe to. Default is all filters.

**CreateRouteTopic()**—Creates route topic.

**Arguments**

- **op**—Operation (**ADD, DELETE, CHANGE, ALL**). Default is **ALL**.
- **family**—Family to subscribe to. Default is all families.
- **prefix\_length**—Default is all prefix lengths.

**CreateRouteTableTopic()**—Creates route table topic.

**Arguments**

- **op**—Operation (**ADD, DELETE, CHANGE, ALL**). Default is **ALL**.
- **table\_name**—Table name to subscribe to. Default is all table names.
- **lr\_name**—LR name to subscribe to. Default is all LR names.

## Programming JET Notification for Non-Python Languages

Many of the high-level languages have an MQTT library available. A JET application can use the corresponding library to connect to MQTT broker running on Junos OS and subscribe for events. For example, please refer the following link for a sample JAVA program connecting to an MQTT broker and subscribing for events: [Example MQTT Messaging in Java](#).

For details on example MQTT libraries for different languages, see [Table 4 on page 19](#).

Table 4: MQTT Libraries by Language

Language Name	License Link
C, Mosquitto	<a href="#">FreeBSD</a>
C++, Mosquitto	<a href="#">FreeBSD</a>
Python, Mosquitto	<a href="#">FreeBSD</a>
Ruby, Ruby-mqtt	<a href="#">MIT License</a>
Java, Eclipse paho Java	<a href="#">Eclipse Public License</a>
Go, Eclipse Paho Go	<a href="#">Eclipse Public License</a>
C#, MqttDotNet	<a href="#">Custom License</a>

**Related  
Documentation**

- [Notification API](#)



## CHAPTER 3

# JET Service APIs

- [JET Service APIs Overview on page 21](#)

## JET Service APIs Overview

---

The JET Service APIs include interfaces in the **JETHandler** Python package that provide access to the control plane on the device and a management interface to run operational and configuration commands.

- [Interfaces Service API on page 21](#)
- [Firewall Service API on page 21](#)

## Interfaces Service API

The [InterfacesService](#) API allows you to add or delete an IPv4 or IPv6 interface.

## Firewall Service API

The [FirewallService](#) API provides the following firewall and traffic policer interfaces:

- Add, change, delete, or replace a firewall filter.
- Bind or unbind a firewall filter to an interface and direction.
- Add, change, or delete a single-rate two-color policer.
- Get statistics associated with a firewall filter counter or policer.
- Clear statistics associated with a firewall filter counter.

**Related Documentation**

- [JET Service APIs](#)

