

Juniper Extension Toolkit API Guide

Published
2020-09-23

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Juniper Extension Toolkit API Guide

Copyright © 2020 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

About the Documentation | v

Documentation and Release Notes | v

Using the Examples in This Manual | v

Merging a Full Example | vi

Merging a Snippet | vii

Documentation Conventions | vii

Documentation Feedback | x

Requesting Technical Support | x

Self-Help Online Tools and Resources | xi

Creating a Service Request with JTAC | xi

1

JET APIs: Release 18.1 and Later

Overview of JET APIs | 13

Notification API | 13

Base Types APIs | 13

Service APIs | 13

Routing | 14

Firewall | 14

Interfaces | 14

Management | 14

JET Notification API Overview | 15

JSON Format of JET Notification Messages | 15

Subscribing to Events | 16

Programming JET Notification for Non-Python Languages | 17

2

JET APIs: Before Release 18.1

JET APIs Reference | 19

JET Release 18.3 | 19

JET Release 18.2 | 20

JET Release 18.1 | 20

JET Release 17.4 | 20

JET Release 17.3 | 21

JET Release 17.2 | 21

JET Release 17.1 | 21

JET Release 16.2 | 22

About the Documentation

IN THIS SECTION

- Documentation and Release Notes | v
- Using the Examples in This Manual | v
- Documentation Conventions | vii
- Documentation Feedback | x
- Requesting Technical Support | x

Use this guide to learn more about Juniper Extension Toolkit (JET) APIs. For more information about developing JET applications, see the *Juniper Extension Toolkit Developer Guide*.

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 <https://www.juniper.net/documentation/>.

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 <https://www.juniper.net/books>.

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xsl;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {  
    file ex-script-snippet.xml; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]  
user@host# edit system scripts  
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]  
user@host# load merge relative /var/tmp/ex-script-snippet.conf  
load complete
```

For more information about the **load** command, see [CLI Explorer](#).

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
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies guide names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS CLI User Guide</i> RFC 1997, <i>BGP Communities Attribute</i>

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

Convention	Description	Examples
<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@# set system domain-name <i>domain-name</i>
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"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	stub <default-metric <i>metric</i> >;
(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.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

GUI Conventions

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

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the [Juniper Networks TechLibrary](#) site, and do one of the following:



- Click the thumbs-up icon if the information on the page was helpful to you.
- Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide 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 Juniper Care or Partner Support Services 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 <https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <https://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: <https://www.juniper.net/customers/support/>
- Search for known bugs: <https://prsearch.juniper.net/>
- Find product documentation: <https://www.juniper.net/documentation/>
- Find solutions and answer questions using our Knowledge Base: <https://kb.juniper.net/>
- Download the latest versions of software and review release notes: <https://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum: <https://www.juniper.net/company/communities/>
- Create a service request online: <https://myjuniper.juniper.net>

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

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit <https://myjuniper.juniper.net>.
- 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 <https://support.juniper.net/support/requesting-support/>.

1

CHAPTER

JET APIs: Release 18.1 and Later

Overview of JET APIs | 13

JET Notification API Overview | 15

Overview of JET APIs

IN THIS SECTION

- [Notification API | 13](#)
- [Base Types APIs | 13](#)
- [Service APIs | 13](#)

JET provides APIs that extend the functionality of Junos OS. These APIs fall into three main categories: the notification API, base type APIs, and service APIs.

Notification API

The notification API provides interfaces that allow you to subscribe to events and designate a callback function to receive events when they occur. See [“JET Notification API Overview” on page 15](#) for more information about this API.

Base Types APIs

If you are using JET Release 18.1 or later, go to [Juniper EngNet](#) for complete documentation of base types APIs. If you are using a JET release before 18.1, go to [“JET APIs Reference” on page 19](#).

- jnx Common Address API
- jnx Common Base Types API

Service APIs

Service APIs provide interfaces to access the control plane on the device and a management interface to run operational and configuration commands. If you are using JET Release 18.1 or later, go to [Juniper EngNet](#) for complete documentation of services APIs. If you are using a JET release before 18.1, go to [“JET APIs Reference” on page 19](#).

Routing

- BFD Service APIs
- BGP Route Service API
- ECMP Tracer Flow Monitoring APIs
- Flexible Tunnel Profile
- Flexible Tunnel Service
- gRIBi
- MPLS Service API
- PRPD Common API
- PRPD Service API
- RIB Service API
- Routing Interface Service API

Firewall

Firewall Service API

Interfaces

Interfaces Service API

Management

- JNX Management Service API
- JNX Authentication Service API
- JNX Registration Service API

JET Notification API Overview

IN THIS SECTION

- [JSON Format of JET Notification Messages | 15](#)
- [Subscribing to Events | 16](#)
- [Programming JET Notification for Non-Python Languages | 17](#)

The JET Notification API allows 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 17). For details about MQTT and Python, see <https://mosquitto.org/>.

JSON Format of JET Notification Messages

JET notification is delivered in JSON format. The JSON message has two parts: the header and the attributes. The header is common for all events. It contains the event ID, hostname, time, severity, and facility of the event. The attributes contain information about the event and 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
    }
  }
}
```

```
}
```

See *Package JET Applications* for more information about JSON.

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"> • /junos/events/kernel/interfaces/ifd/add/ifdname • /junos/events/kernel/interfaces/ifd/change/ifdname • /junos/events/kernel/interfaces/ifd/delete/ifdname 	name, snmp-id, flags
Logical Interface (IFL)	<ul style="list-style-type: none"> • /junos/events/kernel/interfaces/ifl/add/iflname • /junos/events/kernel/interfaces/ifl/change/iflname • /junos/events/kernel/interfaces/ifl/delete/iflname 	name, subunit, snmp-id, flags
Family (IFF)	<ul style="list-style-type: none"> • /junos/events/kernel/interfaces/iff/add/iflname/family-type • /junos/events/kernel/interfaces/iff/change/iflname/family-type • /junos/events/kernel/interfaces/iff/delete/iflname/family-type 	name, subunit, family, table-name, flags
Address	<ul style="list-style-type: none"> • /junos/events/kernel/interfaces/ifa/add/iflname/family-type/address • /junos/events/kernel/interfaces/ifa/change/iflname/family-type/address • /junos/events/kernel/interfaces/ifa/delete/iflname/family-type/address 	name, subunit, family, local-address, destination-address, broadcast-address, flags
Firewall	<ul style="list-style-type: none"> • /junos/events/kernel/firewall/filter/add/filtername • /junos/events/kernel/firewall/filter/change/filtername • /junos/events/kernel/firewall/filter/delete/filtername 	name, version, client-id, filter-type, protocol, interface-name, flags
Route	<ul style="list-style-type: none"> • /junos/events/kernel/route/add/family/prefix-with-length • /junos/events/kernel/route/change/family/prefix-with-length • /junos/events/kernel/route/delete/family/prefix-with-length 	table-name, logical-router-name, address-family, route-type, route-prefix, arrayof(next-hop-address), flags

Table 3: Junos Event Topics (*continued*)

Events	Topic	Event Information Returned
Route-table	<ul style="list-style-type: none"> • /junos/events/kernel/route-table/add/<i>tablename/lrname</i> • /junos/events/kernel/route-table/change/<i>tablename/lrname</i> • /junos/events/kernel/route-table/delete/<i>tablename/lrname</i> 	name, logical-router-name, address-family, flags
Syslog	/junos/events/syslog/ <i>event-id</i>	arrayof(attribute-value pairs)

Programming JET Notification for Non-Python Languages

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

For details on example MQTT libraries for different languages, see [Table 4 on page 17](#).

Table 4: MQTT Libraries by Language

Language Name	License Link
C, Mosquitto	FreeBSD
C++, Mosquitto	FreeBSD
Python, Mosquitto	FreeBSD
Ruby, Ruby-mqtt	MIT License
Java, Eclipse paho Java	Eclipse Public License
Go, Eclipse Paho Go	Eclipse Public License
C#, MqttDotNet	Custom License

RELATED DOCUMENTATION

[Notification API](#)

2

CHAPTER

JET APIs: Before Release 18.1

JET APIs Reference | 19

JET APIs Reference

IN THIS SECTION

- [JET Release 18.3 | 19](#)
- [JET Release 18.2 | 20](#)
- [JET Release 18.1 | 20](#)
- [JET Release 17.4 | 20](#)
- [JET Release 17.3 | 21](#)
- [JET Release 17.2 | 21](#)
- [JET Release 17.1 | 21](#)
- [JET Release 16.2 | 22](#)

If you are using JET Release 18.1 or later, see [“JET Notification API Overview” on page 15](#) for information about the notification API and [Juniper EngNet](#), the Juniper Engineering Network website, for JET API documentation.

Use this reference guide to view documentation for JET APIs before JET Release 18.4.

JET Release 18.3

Authentication Service API	BGP Service API	Interfaces Service API
Firewall Service API	jnx Address API	jnx Base Types API
Management Service API	MPLS Service API	Notification API
PRPD Common API	PRPD Service API	Registration Service API
RIB Service API	Routing Interface Service API	

JET Release 18.2

Authentication Service API	BGP Service API	Interfaces Service API
Firewall Service API	jnx Address API	jnx Base Types API
Management Service API	MPLS Service API	Notification API
PRPD Common API	PRPD Service API	Registration Service API
RIB Service API	Routing Interface Service API	

JET Release 18.1

Authentication Service API	BGP Service API	Interfaces Service API
Firewall Service API	jnx Address API	jnx Base Types API
Management Service API	MPLS Service API	Notification API
PRPD Common API	PRPD Service API	Registration Service API
RIB Service API	Routing Interface Service API	

JET Release 17.4

Authentication Service API	BGP Service API	Class of Service API
Interfaces Service API	Firewall Service API	jnx Address API
jnx Base Types API	Management Service API	MPLS Service API
Notification API	PRPD Common API	PRPD Service API
Registration Service API	RIB Service API	Routing Interface Service API

JET Release 17.3

Authentication Service API	BGP Service API	Class of Service API
Interfaces Service API	Firewall Service API	jnx Address API
jnx Base Types API	Management Service API	MPLS Service API
Notification API	PRPD Common API	PRPD Service API
Registration Service API	RIB Service API	Routing Interface Service API

JET Release 17.2

Authentication Service API	BGP Service API	Class of Service API
Interfaces Service API	Firewall Service API	jnx Address API
Manageability RPC Service API	MPLS Service API	Notification API
PRPD Common API	PRPD Service API	Registration Service API
RIB Service API		

JET Release 17.1

Authentication Service API	BGP Service API	Class of Service API
Interfaces Service API	Firewall Service API	jnx Address API
Manageability RPC Service API	MPLS Service API	Notification API
PRPD Common API	PRPD Service API	Registration Service API
RIB Service API		

JET Release 16.2

Authentication Service API	BGP Service API	DCD Service API
Firewall Service API	jnx Address API	Manageability RPC Service API
Notification API	PRPD Common API	PRPD Service API
Registration Service API		