



Juniper Extension Toolkit Applications Guide



Modified: 2018-07-10

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. 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.

Juniper Extension Toolkit Applications Guide

Copyright © 2018 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://www.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 | vii |
| | Documentation and Release Notes | vii |
| | Using the Examples in This Manual | vii |
| | Merging a Full Example | viii |
| | Merging a Snippet | viii |
| | Documentation Conventions | ix |
| | Documentation Feedback | xi |
| | Requesting Technical Support | xi |
| | Self-Help Online Tools and Resources | xi |
| | Opening a Case with JTAC | xii |
| Chapter 1 | Overview of JET Interaction with Junos OS | 13 |
| | Understanding JET Interaction with Junos OS | 13 |
| | JET Service Process Overview | 14 |
| Chapter 2 | Running a JET Application on a Device Running Junos OS | 15 |
| | Configuring the JET Application and its License on a Device Running Junos OS | 15 |
| | Configuring a Python Application to Run on a Device | 15 |
| | Configuring a C or C++ Application to Run on a Device | 16 |
| | Installing an Application Package on a Device Running Junos OS | 17 |
| | Adding a JET Application Package onto a Device Running Junos OS | 17 |
| | Deploying a JET Application on a Device Running Junos OS | 17 |
| | Enabling a JET Application Configuration on a Device Running Junos OS | 18 |
| | Starting and Stopping a JET Application Running on a Device Running Junos OS | 19 |
| | Verifying Which JET Applications Are Configured | 19 |
| | Starting a JET Application | 20 |
| | Showing the Status of JET Applications on the Device | 20 |
| | Stopping a JET Application | 20 |
| Chapter 3 | Configuring JET Interaction with Junos OS | 23 |
| | Configuring Notification Service for JET Applications | 23 |
| | Configuring Request-Response Service for JET Applications | 24 |
| Chapter 4 | Configuration Statements | 27 |
| | allow-clients | 28 |
| | application (Extensions) | 29 |
| | checksum | 30 |
| | extension-service (System Extensions) | 31 |
| | extension-service (System Services) | 32 |
| | extension-service (System Services gRPC) | 33 |

| | | |
|------------------|--|-----------|
| | extensions | 34 |
| | file (JET) | 36 |
| | grpc | 37 |
| | interface-notification (programmable-rpd) | 38 |
| | language (Scripts) | 39 |
| | local | 40 |
| | max-datasize | 41 |
| | notification | 43 |
| | purge-timeout (routing-options programmable-rpd) | 44 |
| | refresh (JET) | 45 |
| | refresh-from (JET) | 46 |
| | request-response | 47 |
| | routing-instance (JET Scripts) | 48 |
| | services (System Services) | 49 |
| | source (JET Scripts) | 53 |
| | ssl | 54 |
| | traceoptions (Extensions) | 56 |
| | traceoptions (Services) | 58 |
| | traceoptions (routing-options programmable-rpd) | 60 |
| Chapter 5 | Operational Commands | 63 |
| | request extension-service (start stop) | 64 |
| | show extension-service status | 65 |
| | show programmable-rpd clients | 67 |
| | show route | 68 |
| | show route detail | 89 |

List of Tables

| | | |
|------------------|--|------------|
| | About the Documentation | vii |
| | Table 1: Notice Icons | ix |
| | Table 2: Text and Syntax Conventions | ix |
| Chapter 5 | Operational Commands | 63 |
| | Table 3: show extension-service status Output Fields | 65 |
| | Table 4: show route Output Fields | 69 |
| | Table 5: show route detail Output Fields | 89 |
| | Table 6: Next-hop Types Output Field Values | 95 |
| | Table 7: State Output Field Values | 97 |
| | Table 8: Communities Output Field Values | 99 |

About the Documentation

- Documentation and Release Notes on page vii
- Using the Examples in This Manual on page vii
- Documentation Conventions on page ix
- Documentation Feedback on page xi
- Requesting Technical Support on page xi

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.xml;
    }
  }
}
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 ix](#) 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 ix](#) 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 |

Table 2: Text and Syntax Conventions (continued)

| Convention | Description | Examples |
|--------------------------------|---|--|
| Fixed-width text like this | Represents output that appears on the terminal screen. | <pre>user@host> show chassis alarms</pre> <p>No alarms currently active</p> |
| <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> |
| <i>Italic text like this</i> | Represents variables (options for which you substitute a value) in commands or configuration statements. | <p>Configure the machine's domain name:</p> <pre>[edit] root@# set system domain-name domain-name</pre> |
| 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. | <pre>stub <default-metric metric>;</pre> |
| (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. | <pre>broadcast multicast</pre> <p><i>(string1 string2 string3)</i></p> |
| # (pound sign) | Indicates a comment specified on the same line as the configuration statement to which it applies. | <pre>rsvp { # Required for dynamic MPLS only</pre> |
| [] (square brackets) | Encloses a variable for which you can substitute one or more values. | <pre>community name members [community-ids]</pre> |
| Indentation and braces ({ }) | Identifies a level in the configuration hierarchy. | <pre>[edit] routing-options { static { route default { nexthop address; retain; } } }</pre> |
| ;(semicolon) | Identifies a leaf statement at a configuration hierarchy level. | |
| GUI Conventions | | |
| 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. |

Table 2: Text and Syntax Conventions (continued)

| Convention | Description | Examples |
|------------------------------|---|--|
| > (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, 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 <https://www.juniper.net/documentation/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 <https://www.juniper.net/documentation/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 <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/>
- Open a case online in the CSC Case Management tool: <https://www.juniper.net/cm/>

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

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 <https://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 <https://www.juniper.net/support/requesting-support.html>.

CHAPTER 1

Overview of JET Interaction with Junos OS

- [Understanding JET Interaction with Junos OS on page 13](#)
- [JET Service Process Overview on page 14](#)

Understanding JET Interaction with Junos OS

The Juniper Extension Toolkit (JET) provides tools to developers to create applications that extend the functionality of Junos OS. For example, a JET application might extend the Junos CLI by adding a new operational command to show application-specific states.

JET applications can run on devices running Junos OS or run on another type of device in your system and connect over the network to a device running Junos OS.

The JET development environment consists of a virtual machine (VM), an integrated development environment (IDE), and APIs. Click [here](#) to see the rest of the JET documentation.

This guide covers how to deploy and use applications that were developed using JET on Junos OS.

JET applications interact with Junos OS via the following two services:

- **Request-response**—An application issues a request and synchronously waits for the response from Junos OS.
- **Notifications**—An application receives asynchronous notifications of events happening on Junos OS.

The request-response service is included in the JET service process (jsd), which runs on Junos OS. Whenever a request arrives on the TCP port, jsd creates a separate thread to service the JET application request. The session remains established as long as the client and server are both up and able to communicate with each other. Over the lifetime of a session, jsd can execute any number of APIs, and it can execute APIs from multiple sessions in parallel. A maximum of 8 client sessions can be active at any given time.

The notification service uses a publish-subscribe-based messaging protocol and notification broker. JET applications register with the notification broker and inform the broker about the topics for which they are interested in receiving messages. The broker is responsible for distributing messages to the interested clients based on the topic of a

message. Junos OS daemons publishing the events (such as eventd) connect to the broker as a publisher and publish the events.

**Related
Documentation**

- [Configuring Notification Service for JET Applications on page 23](#)
- [Configuring Request-Response Service for JET Applications on page 24](#)
- [JET Service Process Overview on page 14](#)

JET Service Process Overview

To support application interaction with Junos OS, the JET service process (jsd), by default, uses TCP port 32767 to listen for and receive requests from applications to execute APIs. Whenever a request comes on the TCP port, jsd creates a separate thread to service the JET application request. The session remains established as long as the client and server are both up and able to communicate with each other. Over the lifetime of a session, jsd can execute many APIs, and it can execute APIs from multiple sessions in parallel. You can have a maximum of 8 active client sessions connected at any given time.



NOTE: JET does not support ECDSA or DSA SSL certificates. For secure communications with jsd, use RSA certificates, specifically TLSv1.

**Related
Documentation**

- [Understanding JET Interaction with Junos OS on page 13](#)

CHAPTER 2

Running a JET Application on a Device Running Junos OS

- [Configuring the JET Application and its License on a Device Running Junos OS on page 15](#)
- [Installing an Application Package on a Device Running Junos OS on page 17](#)
- [Enabling a JET Application Configuration on a Device Running Junos OS on page 18](#)
- [Starting and Stopping a JET Application Running on a Device Running Junos OS on page 19](#)

Configuring the JET Application and its License on a Device Running Junos OS

Before you can start a JET application on a device running Junos OS, first determine if you must configure the license. License configuration for JET applications is required only if you are deploying on-box applications written in C or C++ and built using the Juniper Extension Toolkit (JET) development environment. For simple Python JET applications, which do not require licensing, this task is not required.

This topic contains two examples of configuring JET applications to run on Junos OS:

- [Configuring a Python Application to Run on a Device on page 15](#)
- [Configuring a C or C++ Application to Run on a Device on page 16](#)

Configuring a Python Application to Run on a Device

To configure a JET Python application and its license on a device:

1. (Optional if Python application is signed) Issue the **set system scripts language python** command.

[edit]

user@device# **set system scripts language python**

If you do not include the **language python** statement, you cannot execute unsigned Python scripts on the device.



NOTE: Junos OS supports using symbolic links for files in the `/var/db/scripts/jet` directory, but the device will only execute the script at the target location if it is signed.

2. At the **[edit system extensions]** hierarchy level, configure the application's provider's ID, for example:

```
[edit system extensions]
user@device# set providers xyzcompany
```



NOTE: The same provider license must be used to configure a JET application to run on Junos OS as was used to package it.

3. Configure the license type and deployment scope.

```
[edit system extensions]
user@device# set providers xyzcompany license-type juniper deployment-scope
commercial
```

4. Commit the configuration.

```
[edit system extensions]
user@device# top
[edit]
user@device# commit
```

Configuring a C or C++ Application to Run on a Device

To configure a JET C or C++ application:

1. Configure the application's provider's ID, license type, and deployment scope.

The following application example was packaged using **chef** as the provider license:

```
[edit]
user@device# set system extensions providers chef license-type juniper
deployment-scope commercial
```



NOTE: The same provider license must be used to configure a JET application to run on Junos OS as was used to package it.

2. Commit the configuration and exit to operational mode.

```
[edit]
user@device# commit
commit complete

[edit]
user@device# exit
user@device>
```

Related Documentation

- *License Modes for Enhanced MPCs Overview*
- *Configuring the License Mode for Specific Enhanced MPCs on MX Series Routers*

- *Software Features That Require Licenses on MX Series Routers Only*

Installing an Application Package on a Device Running Junos OS

After configuring the license of the JET application, you must add the JET application package onto the device running Junos OS. Then you can deploy it.

- [Adding a JET Application Package onto a Device Running Junos OS on page 17](#)
- [Deploying a JET Application on a Device Running Junos OS on page 17](#)

Adding a JET Application Package onto a Device Running Junos OS

To add a JET application package on a device running Junos OS:

- Copy the **application-name.tgz** file to the device running Junos OS, for example:

```
% scp application-name.tgz device-hostname:/var/tmp
```

You can use other copy commands or FTP for this step as well.

Deploying a JET Application on a Device Running Junos OS

Before you begin this procedure, add the JET application package onto the device. Then, go into the operational mode of the CLI.

To deploy the JET application:

1. Issue the **request system software add** operational command from the CLI.

```
user@device> request system software add application.tgz
```

The package validates the certificate with the configured provider ID. The provider ID is a uniquely identifying prefix that represents the name of the developer's organization. If the provider ID matches that on the package, the package is deployed.

Using the **request system software add** CLI command brings the application into the **/var/db/scripts/jet** directory automatically.

Following is an example:

```
user@device> request system software add
/var/tmp/echoclient-16.1I20160406_0623_root.tgz
Removing package 'echoclient' ...
Unmounted /packages/mnt/echoclient-16.1I20160406_0623_root ...
Installing package '/var/tmp/echoclient-16.1I20160406_0623_root.tgz' ...
Verified echoclient-16.1I20160406_0623_root signed by chef-juniper-commercial-1
method RSA2048+SHA1 chef
Mounted echoclient package on /dev/md20...
Verified manifest signed by chef-juniper-commercial-1 method RSA2048+SHA1
Saving package file in /var/sw/pkg/echoclient-16.1I20160406_0623_root.tgz ...
Saving state for rollback ...
The app is installed:

user@device> request system software add
/var/tmp/echoserver-16.1I20160404_0845_root.tgz
```

```
Removing package 'echoserver' ' ' ...
Unmounted /packages/mnt/echoserver-16.1I20160404_0845_root ...
Installing package '/var/tmp/echoserver-16.1I20160404_0845_root.tgz' ...
Verified echoserver-16.1I20160404_0845_root signed by chef-juniper-commercial-1
  method RSA2048+SHA1 chef
Mounted echoserver package on /dev/md21...
Verified manifest signed by chef-juniper-commercial-1 method RSA2048+SHA1
Saving package file in /var/sw/pkg/echoclient-16.1I20160406_0623_root.tgz ...
Saving state for rollback ...
The app is installed:
```

The daemons echoclientd and echoserverd are put in the `/var/db/scripts/jet` path automatically.

2. Verify the version of your application.

```
user@device> show version
JET echoclient example Application [16.2I20161106_0623_root]
JET echoserver example Application [16.2I20161104_0845_root]
```

Related Documentation

- [Configuring the JET Application and its License on a Device Running Junos OS on page 15](#)
- [Enabling a JET Application Configuration on a Device Running Junos OS on page 18](#)

Enabling a JET Application Configuration on a Device Running Junos OS

The configuration described in this topic is required to enable the JET application to run on a device running Junos OS.

To enable a signed JET application to run on a device running Junos OS:

1. Go to the `[edit system extensions extension-service application]` CLI hierarchy level.

```
[edit]
user@device# edit system extensions extension-service application
```

2. Configure the application script name.

```
[edit system extensions extension-service application]
user@device# set file script-name
```

3. (Optional) Configure the **arguments** and **daemonize** statements.

Use the **arguments** statement when you need to pass arguments to the application.

Use the **daemonize** statement when you want the application to run as a background process. This means that once committed, the daemonize application is started automatically, with no manual intervention required.

```
[edit system extensions extension-service application script-name]
user@device# set arguments [ argument1 argument2 ] daemonize
```

4. (Optional) Configure the **username** statement.

Use the **username** statement to set a user name under which the script will run. This allows the script to run with the privileges of the specified user. If not set, the script runs with the privileges of the nobody user.

```
[edit system extensions extension-service application script-name]  
user@device# set username username
```

5. Commit.

```
[edit system extensions extension-service application]  
user@device# top  
[edit]  
user@device# commit
```

Once you commit, you will have the following configuration:

```
user@device# show system extensions extension-service application  
file script-name {  
  arguments [ argument1 argument2 ];  
  daemonize;  
}
```

**Related
Documentation**

- [file \(JET\) on page 36](#)
- [Installing an Application Package on a Device Running Junos OS on page 17](#)
- [Configuring the JET Application and its License on a Device Running Junos OS on page 15](#)

Starting and Stopping a JET Application Running on a Device Running Junos OS

This topic describes how to run a JET application on a device running Junos OS and how to show its status.

- [Verifying Which JET Applications Are Configured on page 19](#)
- [Starting a JET Application on page 20](#)
- [Showing the Status of JET Applications on the Device on page 20](#)
- [Stopping a JET Application on page 20](#)

Verifying Which JET Applications Are Configured

To verify which JET applications are configured to run:

- Use the **show system extensions** configuration mode command.

```
user@device# show system extensions
```

For example, suppose echoclient and echoserver have been installed and enabled on a device. The output from the **show system extensions** command might be the following:

```
providers {
  chef {
    license-type juniper deployment-scope commercial;
  }
}
extension-service {
  application {
    file echoclientd {
      arguments "192.0.2.1 echo-from-client";
    }
    file echoserverd {
      daemonize;
    }
  }
}
```

Starting a JET Application

To start a JET application:

1. Issue the **request extension-service start *application-name*** operational command, for example:

```
user@device> request extension-service start echoclientd
Extension-service application 'echoclientd' started with pid: 85280
-- server reply:echo-from-client
```

Showing the Status of JET Applications on the Device

To show the status of the JET applications on a device running Junos OS:

- Use the **show extension-service status** command:



NOTE: The **show extension-service status** operational command is limited to use with Python applications only.

- To display the status of one application, use the **show extension-service status *application-name*** command.

```
user@device> show extension-service status application-name
```

- To display the status of multiple applications, use the **show extension-service status all** command.

```
user@device> show extension-service status all
```

Stopping a JET Application

To stop the application:

- Issue the **request extension-service stop** command.

```
user@device> request extension-service stop application-name
```

- Related Documentation**
- [show extension-service status on page 65](#)
 - [request extension-service \(start | stop\) on page 64](#)
 - [Enabling a JET Application Configuration on a Device Running Junos OS on page 18](#)

CHAPTER 3

Configuring JET Interaction with Junos OS

- [Configuring Notification Service for JET Applications on page 23](#)
- [Configuring Request-Response Service for JET Applications on page 24](#)

Configuring Notification Service for JET Applications

In notification service, the application receives asynchronous notifications of events occurring on Junos OS.

Junos OS events are published with topics. Clients can subscribe to events they are interested in by using topics. Once a client subscribes, only the events matching the topic or topics are delivered to the client. This type of messaging requires a *broker*. The message broker is responsible for distributing messages to the subscribed clients based on the topic of a message.

The default behavior is to disallow all clients to connect to the broker. In order to allow clients to connect to the broker for notifications, users must configure the CLI **allow-clients** statement at the **[edit system services extension-service notification]** hierarchy level.

To configure a device to support notifications for JET applications:

1. Enter the CLI and go to the **[edit system services extension-service notification]** hierarchy level.

```
% cli
user@device> configure
```

```
[edit]
user@device# edit system services extension-service notification
```

```
[edit system services extension-service notification]
```

2. Configure the **allow-clients** statement.

```
user@device# set allow-clients address ip-address
```

You can set a series of addresses at once:

```
user@device# set allow-clients address [address1 address2 address3]
```

3. Configure the maximum connections.

4. Configure the port number.

Following is the hierarchy for the **system services extension-service notification** configuration statement:

```
notification {
  allow-clients { /* Clients for which notification session requests are allowed */
    address ip-address /* Default: None */
  }
  max-connections number; /* Default: 20 */
  port number; /* Port number to accept incoming connections. Default: 1883, which is
    used for communication in plain text */
}
```

**Related
Documentation**

- [allow-clients on page 28](#)
- [Configuring Request-Response Service for JET Applications on page 24](#)
- [Understanding JET Interaction with Junos OS on page 13](#)

Configuring Request-Response Service for JET Applications

When using the request-response service, the client application issues a request and synchronously waits for the response from the Junos OS server.

You can configure the JET service process (jsd) to run in clear text or Secure Sockets Layer (SSL) mode for increased security. To have jsd run in SSL mode, you must first enable the jsd process to use SSL by adding and configuring the certificate name locally. The certificate must be an RSA certificate. ECDSA and DSA SSL certificates are not supported.

Currently, JET supports Transport Layer Security (TLS) version 1.2 for certificate exchange and supports multiple encryption algorithms, but does not support mutual authentication. This means that clients can authenticate the server, but the server can not authenticate clients using SSL / TLS certificates. For client authentication, use the LoginCheck() procedure from the authentication service API.

To configure the jsd for request-response service:

1. Copy the SSL certificate and private key PEM file to the device using the FTP command line or the **scp** command.

For example, if **ssl-jsd-encrypt.pem** is the SSL certificate's filename:

```
% scp ssl-jsd-encrypt.pem device-name:/var/tmp
```

2. Import the SSL certificate and private key with the CLI **set security certificates local *local-name* load-key-file *pathname-on-device*** configuration statement.

For example, if the local name of the SSL certificate is `jsd_certificate`:

```
[edit]
user@device# set security certificates local jsd_certificate load-key-file
/var/tmp/ssl-jsd-encrypt.pem
```

3. Go to the `[edit system services extension-service request-response grpc]` hierarchy level.

```
[edit]
user@device# edit system services extension-service request-response grpc
[edit system services extension-service request-response grpc]
```

4. Specify either `clear-text` or `ssl`.



NOTE: Setting `ssl` requires that you must first enable the `jsd` process to use SSL. See Step 1.

5. Specify the maximum connections.

6. Specify the scripts to use.

Following is the hierarchy for the `system services extension-service request-response grpc` configuration statement:

```
clear-text { /* Enables clear-text-based API connections. Set either this or ssl,
but not both. */
  address ip-address; /* Default: :: */
  port port; /* Default: 32767 */
}
max-connections max-connections; /* Default: 5 */
ssl { /* Enables SSL-based API connections. */
  address ip-address; /* Default: :: */
  local-certificate local-certificate;
  port port; /* Default: 32767 */
}
```



NOTE: gRPC identifies IP addresses as strings that can represent an IPv4 address, IPv6 address, or a host name. If you configure the address values as “::”, Junos OS converts the address internally so that gRPC listens on a wildcard socket like `tcp46 0 0 *:50051`. If a client connects using an IPv4 address, then the connection becomes an IPv4 mapped IPv6 socket at the TCP level.

Following is more information about the options for the **system services extension-service request-response grpc** configuration statement:

- For SSL, there is no default for **local-certificate**. The value for **local-certificate** should be the same as the name provided during the import of the certificate using the CLI configuration statement **local** in the **[edit security certificates]** hierarchy level. In our example, the local certificate name is `jsd_certificate`.
- **max-connections** is the number of simultaneous connections for request-response that can be attached to `jsd`. The higher the number, the higher the impact that clients have on performance. The maximum number of connections supported is 8.

**Related
Documentation**

- [local on page 40](#)
- [language \(Scripts\) on page 39](#)
- [Configuring Notification Service for JET Applications on page 23](#)
- [Understanding JET Interaction with Junos OS on page 13](#)

CHAPTER 4

Configuration Statements

- [allow-clients](#) on page 28
- [application](#) (Extensions) on page 29
- [checksum](#) on page 30
- [extension-service](#) (System Extensions) on page 31
- [extension-service](#) (System Services) on page 32
- [extension-service](#) (System Services gRPC) on page 33
- [extensions](#) on page 34
- [file](#) (JET) on page 36
- [grpc](#) on page 37
- [interface-notification](#) (programmable-rpd) on page 38
- [language](#) (Scripts) on page 39
- [local](#) on page 40
- [max-datasize](#) on page 41
- [notification](#) on page 43
- [purge-timeout](#) (routing-options programmable-rpd) on page 44
- [refresh](#) (JET) on page 45
- [refresh-from](#) (JET) on page 46
- [request-response](#) on page 47
- [routing-instance](#) (JET Scripts) on page 48
- [services](#) (System Services) on page 49
- [source](#) (JET Scripts) on page 53
- [ssl](#) on page 54
- [traceoptions](#) (Extensions) on page 56
- [traceoptions](#) (Services) on page 58
- [traceoptions](#) (routing-options programmable-rpd) on page 60

allow-clients

| | |
|---------------------------------|---|
| Syntax | <pre>allow-clients { address <i>ip-address</i>; }</pre> |
| Hierarchy Level | [edit system services extension-service notification] |
| Release Information | Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | Specify client IP addresses from which notifications are allowed. |
| Options | address <i>ip-address</i> —Specify IPv4 or IPv6 addresses (prefix length optional) or host names. You can specify a set of values using square brackets ([]). |
| Required Privilege Level | system—To view this statement in the configuration. system-control—To add this statement to the configuration. |
| Related Documentation | <ul style="list-style-type: none">• notification on page 43 |

application (Extensions)

| | |
|---------------------------------|---|
| Syntax | <pre> application { file <i>script-name</i> { arguments <i>arguments</i>; checksum <i>hash-algorithm</i> <i>hash</i>; daemonize; username <i>username</i>; } max-datasize <i>max-datasize</i>; traceoptions { file <<i>filename</i>> <files <i>number</i>> <size <i>size</i>> <world-readable no-world-readable>; flag <i>flag</i>; no-remote-trace; } } </pre> |
| Hierarchy Level | [edit system extensions extension-service] |
| Release Information | Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | <p>Configure the Junos OS extension service application.</p> <p>The remaining statements are explained separately. See CLI Explorer.</p> |
| Required Privilege Level | <p>maintenance—To view this statement in the configuration.</p> <p>maintenance-control—To add this statement to the configuration.</p> |
| Related Documentation | <ul style="list-style-type: none"> extension-service (System Extensions) on page 31 |

checksum

| | |
|---------------------------------|---|
| Syntax | <code>checksum (md5 sha-256 sha1) <i>hash</i>;</code> |
| Hierarchy Level | [edit event-options event-script file <i>filename</i>], [edit system scripts commit file <i>filename</i>], [edit system scripts op file <i>filename</i>], [edit system scripts snmp file <i>filename</i>], [edit system extensions extension-service application file <i>filename</i>] |
| Release Information | Statement introduced in Junos OS Release 9.5. Statement introduced in Junos OS Release 11.1 for the QFX Series. Statement introduced in Junos OS Release 13.2X51-D10 for QFX Series switches. Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series. Statement introduced for the [edit system extensions extension-service application file <i>filename</i>] hierarchy level in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | For Junos OS commit scripts, event scripts, op scripts, SNMP scripts, and scripts developed using the Juniper Extension Toolkit (JET) specify the MD5, SHA-1, or SHA-256 checksum hash. When Junos OS executes a local commit, event, op, or SNMP script, the system verifies the authenticity of the script by using the configured checksum hash. |
| Options | md5 <i>hash</i> —MD5 checksum of this script. sha-256 <i>hash</i> —SHA-256 checksum of this script. sha1 <i>hash</i> —SHA-1 checksum of this script. |
| Required Privilege Level | maintenance —To view this statement in the configuration. maintenance-control —To add this statement to the configuration. |
| Related Documentation | <ul style="list-style-type: none">• <i>Configuring Checksum Hashes for a Commit Script</i>• <i>Configuring Checksum Hashes for an Event Script</i>• <i>Configuring Checksum Hashes for an Op Script</i>• <i>Configuring Checksum Hashes for an SNMP Script</i>• <i>Executing an Op Script from a Remote Site</i>• <i>file checksum md5</i>• <i>file checksum sha-256</i>• <i>file checksum sha1</i> |

extension-service (System Extensions)

| | |
|---------------------------------|--|
| Syntax | <pre> extension-service { application { file script-name { arguments arguments; checksum hash-algorithm hash; daemonize; refresh; refresh-from; respawn-on-normal-exit; routing-instance source; username username; } max-datasize max-datasize; traceoptions { file <filename> <files number> <size size> <world-readable no-world-readable>; flag flag; no-remote-trace; } } } </pre> |
| Hierarchy Level | [edit system extensions] |
| Release Information | <p>Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.</p> <p>Statement introduced in Junos OS Release 17.1R1 for all MX-Series routers.</p> <p>Statement introduced in Junos OS Release 17.1R1 for ACX500, ACX1000, ACX1100, ACX2100, ACX2200, ACX4000 routers.</p> <p>Statement introduced in Junos OS Release 17.2R1 for PTX-Series routers.</p> <p>Statement introduced in Junos OS Release 17.3R1 for SRX-Series Services Gateways.</p> <p>Argument respawn-on-normal-exit introduced in Junos OS Releases 17.3R3 and (Commented out until release of 17.3R3 on 4/19/2018) 18.1R1.</p> |
| Description | <p>Enable Junos OS extension services.</p> <p>The refresh option instructs the system to refresh all jet applications from their source.</p> <p>The refresh-from option instructs the system to refresh all jet applications from a given base URL.</p> <p>The source option provides the specific URL of the source for this application.</p> <p>The remaining statements are explained separately. See CLI Explorer.</p> |
| Required Privilege Level | <p>maintenance—To view this statement in the configuration.</p> <p>maintenance-control—To add this statement to the configuration.</p> |

- Related Documentation**
- [Enabling a JET Application Configuration on a Device Running Junos OS on page 18](#)
 - [extensions on page 34](#)

extension-service (System Services)

Syntax

```
extension-service {
  request-response {
    grpc {
      (clear-text | ssl) {
        address ip-address;
        local-certificate local-certificate; # Not specified for clear-text option
        port port;
      }
      max-connections max-connections;
    }
  }
  notification {
    port port;
    max-connections max-connections;
    allow-clients {
      address ip-address;
    }
  }
  traceoptions {
    file <filename> <files number> <match regex> <size size> <world-readable |
    no-world-readable>;
    flag flag;
    no-remote-trace;
  }
}
```

Hierarchy Level [edit system services]

Release Information Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.

Description Enable Junos OS extension services.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

- Related Documentation**
- [services \(System Services\) on page 49](#)

extension-service (System Services gRPC)

| | |
|---------------------------------|--|
| Syntax | <pre> extension-service { request-response { grpc { (clear-text ssl) { address <i>ip-address</i>; local-certificate <i>local-certificate</i>; # Not specified for clear-text option port <i>port</i>; } max-connections <i>max-connections</i>; } } notification { port <i>port</i>; } traceoptions { file <<i>filename</i>>; flag <i>flag</i>; flag <i>flag</i>; } } </pre> |
| Hierarchy Level | [edit system services] |
| Release Information | Statement introduced in Junos OS Release 16.2 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | Enable Junos OS extension services. |
| Required Privilege Level | system—To view this statement in the configuration. system-control—To add this statement to the configuration. |
| Related Documentation | <ul style="list-style-type: none"> • services (System Services) on page 49 • grpc on page 37 |

extensions

```
Syntax extensions {
    extension-service {
        application {
            file script-name {
                arguments arguments;
                checksum hash-algorithm hash;
                daemonize;
                username username;
            }
            max-datasize max-datasize;
            traceoptions {
                file <filename> <files number> <size size> <world-readable | no-world-readable>;
                flag flag;
                no-remote-trace;
            }
        }
    }
    providers {
        provider-id {
            license-type license deployment-scope [ deployments ];
        }
    }
    resource-limits {
        package package-name {
            resources {
                cpu {
                    priority number;
                    time seconds;
                }
                file {
                    core-size bytes;
                    open number;
                    size bytes;
                }
                memory {
                    data-size bytes;
                    locked-in bytes;
                    resident-set-size bytes;
                    socket-buffers bytes;
                    stack-size bytes;
                }
            }
        }
    }
    process process-ui-name {
        resources {
            cpu {
                priority number;
                time seconds;
            }
            file {
                core-size bytes;
                open number;
            }
        }
    }
}
```

```

        size bytes;
    }
    memory {
        data-size bytes;
        locked-in bytes;
        resident-set-size bytes;
        socket-buffers bytes;
        stack-size bytes;
    }
}
}
}
}

```

| | |
|---------------------------------|---|
| Hierarchy Level | [edit system] |
| Release Information | Statement introduced in Junos OS Release 9.0. extension-service option introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | <p>Configure extensions to Junos OS.</p> <p>You must configure the providers <i>provider-id</i> statement to enable application packages developed using the Junos SDK to be deployed and run on the router.</p> <p>You must configure the extension-service statement to enable application packages developed using the Juniper Extension Toolkit (JET) to be deployed and run on the device.</p> <p>The remaining statements are explained separately. See CLI Explorer.</p> |
| Required Privilege Level | <p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p> |
| Related Documentation | <ul style="list-style-type: none"> • Installing Application Packages • Enabling a JET Application Configuration on a Device Running Junos OS on page 18 |

file (JET)

Syntax file *filename* {
 arguments *arguments*;
 checksum *hash-algorithm hash*;
 daemonize;
 refresh;
 refresh-from;
 routing-instance;
 source;
 username *username*;
 }

Hierarchy Level [edit system extensions extension-service [application](#)]

Release Information Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.
 refresh, **refresh-from**, **routing-instance**, and **source** options added in Junos OS Release 18.1R1 for MX Series, PTX Series, and QFX Series.

Description For files in the [edit system extensions extension-service [application](#)] hierarchy level, specify the configuration for each file in the extension-service application.

Options **arguments** *arguments*—Specify the command-line arguments called by a JET application.
 A program can take any number of command-line arguments. Enter the arguments in the way the application expects. Developer must supply this information.

daemonize—Specify the file as daemonized.

An application runs as a daemonized process in the background. An application configured to run as a daemonized process is automatically triggered upon commit. A non-daemonized application must be triggered manually from the command-line client.

filename—Local filename of the script file.

refresh—Refresh all operation scripts from their source

refresh-from—Refresh all operation scripts from a given base URL

source—URL source used for refresh for this script

username *username*—Specify the name of the user under whose privileges the extension service will execute. This username is configured at the [edit system login] hierarchy level. If you do not associate a username with an extension-service application, the application is executed as user nobody.

Default: nobody

The remaining statement is explained separately. See [CLI Explorer](#).

| | |
|---------------------------|---|
| Required Privilege | maintenance—To view this statement in the configuration. |
| Level | maintenance-control—To add this statement to the configuration. |


grpc

| | |
|------------------------------|--|
| Syntax | <pre>grpc { clear-text { address <i>ip-address</i>; port <i>port</i>; } }</pre> |
| Hierarchy Level | [edit system services extension-service request-response] |
| Release Information | Statement introduced in Junos OS Release 16.2 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | Configure the type of connections the gRPC service accepts for API applications. |
| Required Privilege | system—To view this statement in the configuration. |
| Level | system-control—To add this statement to the configuration. |
| Related Documentation | <ul style="list-style-type: none"> • extension-service on page 33 • request-response on page 47 • Configuring Request-Response Service for JET Applications on page 24 • JET Service Process Overview on page 14 |


interface-notification (programmable-rpd)

| | |
|---------------------------------|---|
| Syntax | interface-notification <i>name</i> ; |
| Hierarchy Level | [edit logical-systems <i>name</i> routing-instances <i>name</i> routing-options programmable-rpd client <i>id</i>], [edit logical-systems <i>name</i> routing-options programmable-rpd client <i>id</i>], [edit routing-instances <i>name</i> routing-options programmable-rpd client <i>id</i>], [edit routing-options programmable-rpd client <i>id</i>] |
| Release Information | Statement introduced in Junos OS Release 17.4R1 |
| Description | Restrict interface event notifications from the programmable routing protocol process (prpd) to specified JET clients and interfaces. The prpd provides public APIs to program routing systems, making it possible for users to directly access the APIs to customize, create and modify behavior of their network. |
| Default | No restrictions are imposed by default and JET clients are notified of all interfaces. |
| Options | name —Interface name |
| Required Privilege Level | routing |
| Related Documentation | <ul style="list-style-type: none">• show programmable-rpd clients on page 67• traceoptions (routing-options programmable-rpd) on page 60• purge-timeout (routing-options programmable-rpd) on page 44 |


language (Scripts)

| | |
|---------------------------------|---|
| Syntax | language python; |
| Hierarchy Level | [edit system scripts] |
| Release Information | <p>Statement introduced in Junos OS Release 16.1 on QFX Series switches and MX Series, PTX Series, and T Series routers.</p> <p>Statement introduced in Junos OS Release 17.1 on EX Series switches.</p> <p>Statement introduced in Junos OS Release 17.3 on SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances.</p> |
| Description | <p>Enable unsigned Python scripts to be executed on a device running Junos OS.</p> <p>Configuring the language python statement enables you to execute unsigned Python commit, event, op, and SNMP automation scripts as well as execute unsigned Python scripts developed using the Juniper Extension Toolkit (JET) on devices running Junos OS. To prevent the execution of unauthorized Python code, unsigned Python scripts must be owned by either the root user or a user in the Junos OS super-user login class, and only the file owner can have write permission for the file.</p> <hr/> <div>  <p>NOTE: Starting in Junos OS Release 16.1R3, unsigned Python scripts must be owned by either root or a user in the Junos OS super-user login class, and only the file owner can have write permission for the file. Prior to Junos OS Release 16.1R3, unsigned Python scripts must be owned by the root user.</p> </div> <hr/> <p>Python commit, event, op, and SNMP scripts must be configured under the hierarchy level appropriate to the script type, and the filename must include the .py extension. To execute Python op scripts from a remote site, you must also configure the allow-url-for-python statement at the [edit system scripts op] hierarchy level.</p> |
| Default | If you do not include the language python statement, you cannot execute unsigned Python scripts on the device. |
| Required Privilege Level | <p>maintenance—To view this statement in the configuration.</p> <p>maintenance-control—To add this statement to the configuration.</p> |
| Related Documentation | <ul style="list-style-type: none"> • <i>Understanding Python Automation Scripts for Devices Running Junos OS</i> • Configuring the JET Application and its License on a Device Running Junos OS on page 15 • file (JET) on page 36 |

local

| | |
|--|--|
| Syntax | <pre>local <i>certificate-name</i> { <i>certificate-key-string</i>; load-key-file <i>URL filename</i>; }</pre> |
| Hierarchy Level | [edit security certificates] |
| Release Information | Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 11.1 for the QFX Series. Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series. |
| Description | Import a paired X.509 private key and authentication certificate, to enable Junos XML protocol client applications to establish Secure Sockets Layer (SSL) connections to the router or switch. |
| <div> NOTE: For FIPS mode, the digital security certificates must be compliant with the National Institute of Standards and Technology (NIST) SP 800-131A standard.</div> | |
| Options | <p><i>certificate-key-string</i>—String of alphanumeric characters that constitute the private key and certificate.</p> <p><i>certificate-name</i>—Name that uniquely identifies the certificate.</p> <p><i>load-key-file URL filename</i>—File that contains the private key and certificate. It can be one of two types of values:</p> <ul style="list-style-type: none">• Pathname of a file on the local disk (assuming you have already used another method to copy the certificate file to the router's or switch's local disk)• URL to the certificate file location (for instance, on the computer where the Junos XML protocol client application runs) |
| Required Privilege Level | system—To view this statement in the configuration. system-control—To add this statement to the configuration. |
| Related Documentation | <ul style="list-style-type: none">• <i>Importing SSL Certificates for Junos XML Protocol Support</i> |

max-datasize

| | |
|--|--|
| Syntax | <code>max-datasize size;</code> |
| Hierarchy Level | <p>[edit event-options event-script], [edit system extension extension-service application], [edit system scripts commit], [edit system scripts op], [edit system scripts snmp], [edit system scripts translation]</p> |
| Release Information | <p>Statement introduced in Junos OS Release 12.3.</p> <p>Statement introduced in Junos OS Release 13.2X51-D10 for QFX Series switches.</p> <p>Support at the [edit system extension extension-service application] hierarchy level introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, and vMX.</p> <p>Support at the [edit system scripts translation] hierarchy level introduced in Junos OS Release 16.1.</p> |
| Description | <p>Maximum amount of memory allocated for the data segment during execution of a script of the configured type. Junos OS sets the maximum memory limit for the executing script to the configured value irrespective of the total memory available on the system at the time of execution. If the executing script exceeds the specified maximum memory limit for that script type, it exits gracefully.</p> |
| <div>  <p>NOTE: For op scripts, the <code>max-datasize</code> statement is only enforced for op scripts that are local to the device. If you execute an op script from a remote location using the <code>op url</code> command, Junos OS uses the default memory allocation settings.</p> </div> | |
| Default | <p>If you do not include the <code>max-datasize</code> statement, the system allocates half of the total available memory of the system up to a maximum value of 128 MB for the data segment portion of the executed script.</p> |
| Options | <p>size—Maximum amount of memory allocated for the data segment during execution of a script of the given type. If you do not specify a unit of measure, the default is bytes.</p> <p>Syntax: size to specify bytes, sizek to specify KB, sizem to specify MB, or sizeg to specify GB</p> <p>Range:</p> <ul style="list-style-type: none"> 32-bit Junos OS—23,068,672 bytes (22 MB) through 1,073,741,824 bytes (1 GB) 64-bit Junos OS—23,068,672 bytes (22 MB) through 1,073,741,824 bytes (1 GB) (SNMP scripts) |

- 64-bit Junos OS—23,068,672 bytes (22 MB) through 3,221,225,472 bytes (3 GB) (commit, event, op, translation, and extension service scripts)



NOTE: The maximum memory for extension service scripts in 64-bit Junos OS images is 3,221,225,472 bytes (3 GB) starting in Junos OS Releases 16.1R4, 16.2R2, and 17.1R1. Prior to these releases, the maximum is 1,073,741,824 bytes (1 GB).

Required Privilege Level maintenance—To view this statement in the configuration.
maintenance-control—To add this statement to the configuration.

Related Documentation

- *max-policies*
- *Understanding Limits on Executed Event Policies and Memory Allocation for Scripts*
- *Example: Configuring Limits on Executed Event Policies and Memory Allocation for Scripts*


notification

| | |
|---------------------------------|--|
| Syntax | <pre>notification { port <i>port</i>; max-connections <i>max-connections</i>; allow-clients { address <i>ip-address</i>; } }</pre> |
| Hierarchy Level | [edit system services extension-service] |
| Release Information | Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | Enable notification services for applications running on devices running Junos OS. |
| Options | <p>max-connections <i>max-connections</i>—Specify the maximum number of connections. Range: 1 through 20 Default: 20</p> <p>port <i>port</i>—Specify the number of the port to accept incoming connections. Range: 1 through 65535 Default: 1883</p> <p>The remaining statements are explained separately. See CLI Explorer.</p> |
| Required Privilege Level | <p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p> |
| Related Documentation | <ul style="list-style-type: none"> • Configuring Notification Service for JET Applications on page 23 • Understanding JET Interaction with Junos OS on page 13 • extension-service on page 32 |


purge-timeout (routing-options programmable-rpd)

| | |
|---------------------------------|---|
| Syntax | <code>purge-timeout <seconds></code> |
| Hierarchy Level | [edit routing-options programmable-rpd] |
| Release Information | Statement introduced in Junos OS Release 16.2. |
| Description | <p>Set the time, in seconds, after which a disconnected client times-out. Upon disconnect, the client state remain available but no operations occur. If the disconnected client reconnects before the set time has elapsed, the states are restored on the router. If it does not, all client operations are reverted and the programmable routing protocol process (prpd) server notifies any other modules of the disconnect so they can purge any other client operations.</p> <p>The prpd provides public APIs to program routing systems, making it possible for users to directly access the APIs to customize, create and modify behavior of their network.</p> |
| Options | <p>Values:</p> <p>purge-timeout <i>seconds</i> —(Optional) Set the time, in seconds, after which disconnected clients time-out on the PRPD server.</p> <p>Range: 1 through 1000</p> <p>Default: 120</p> |
| Required Privilege Level | <p>routing and trace—To view this statement in the configuration.</p> <p>routing-control and trace-control—To add this statement to the configuration.</p> |
| Related Documentation | <ul style="list-style-type: none">• traceoptions on page 60• show programmable-rpd clients on page 67• show route on page 68 |

refresh (JET)

| | |
|---------------------------------|---|
| Syntax | <code>refresh;</code> |
| Hierarchy Level | [edit system extensions extension-service application file filename] |
| Release Information | Statement introduced in Junos OS Release 18.1R1. |
| Description | <p>Overwrite the local copy of all enabled commit scripts or a single enabled commit script with the copy located at the source URL, as specified in the source statement at the same hierarchy level. If the load-scripts-from-flash statement is configured, the device refreshes the scripts on the flash drive instead of the hard disk.</p> <p>The update operation occurs as soon as you issue the set refresh configuration mode command. Issuing the set refresh command does not add the refresh statement to the configuration. Thus the command behaves like an operational mode command by executing an operation, instead of adding a statement to the configuration.</p> |
| | <p> NOTE: On the QFabric system, commit scripts are stored in the <code>/pbdata/mgd_shared/partition-ip/var/db/scripts/commit/</code> directory on the Director device.</p> |
| | <p>As of Junos OS Release 18.1R1, you can specify which routing instance the update is done through. To specify the routing instance to use for updating commit scripts, configure the routing instance in two places in the CLI:</p> <pre> user@host# set system routing-instances <i>routing-instance-name</i> description <i>description</i> user@host# set system extensions extension-service application file <i>filename</i> routing-instance <i>routing-instance-name</i> </pre> <p>If you enable the non-default management instance and use mgmt_junos for routing-instance-name, you can configure scripts to update using the dedicated management instance mgmt_junos.</p> |
| Required Privilege Level | <p>maintenance—To view this statement in the configuration.</p> <p>maintenance-control—To add this statement to the configuration.</p> |
| Related Documentation | <ul style="list-style-type: none"> • <i>Configuring and Using a Master Source Location for a Script</i> • <i>Example: Configuring and Refreshing from the Master Source for a Script</i> • refresh-from (JET) on page 46 • source (JET Scripts) on page 53 • routing-instance (JET Scripts) on page 48 |


refresh-from (JET)

| | |
|---------------------------------|---|
| Syntax | <code>refresh-from url;</code> |
| Hierarchy Level | [edit system extensions extension-service application file filename] |
| Release Information | Statement introduced in Junos OS Release 18.1R1. |
| Description | <p>Overwrite the local copy of all enabled commit scripts or a single enabled commit script with the copy located at the specified URL. If the load-scripts-from-flash statement is configured, the device refreshes the scripts on the flash drive instead of the hard disk.</p> <p>The update operation occurs as soon as you issue the set refresh-from url configuration mode command. Issuing the set refresh-from command does not add the refresh-from statement to the configuration. Thus the command behaves like an operational mode command by executing an operation, instead of adding a statement to the configuration.</p> |
| | <hr/> <div> NOTE: This statement is not supported on the QFabric system.</div> <hr/> |
| | <p>As of Junos OS Release 18.1R1, you can specify which routing instance the update is done through. To specify the routing instance to use for updating op scripts, configure the routing instance in two places in the CLI:</p> <pre>user@host# set system routing-instances <i>routing-instance-name</i> description <i>description</i> user@host# set system extensions extension-service application file <i>filename</i> routing-instance <i>routing-instance-name</i></pre> <p>If you enable the non-default management instance and use mgmt_junos for routing-instance-name, you can configure scripts to update using the dedicated management instance mgmt_junos.</p> |
| Options | url —The source specified as a Hypertext Transfer Protocol (HTTP) URL, FTP URL, or secure copy (scp)-style remote file specification. |
| Required Privilege Level | maintenance —To view this statement in the configuration. maintenance-control —To add this statement to the configuration. |
| Related Documentation | <ul style="list-style-type: none">• Using an Alternate Source Location for a Script• refresh (JET) on page 45• source (JET Scripts) on page 53• routing-instance (JET Scripts) on page 48 |

request-response

| | |
|---------------------------------|--|
| Syntax | <pre>request-response { grpc { max-connections <i>max-connections</i>; ssl { address <i>ip-address</i>; local-certificate <i>local-certificate-information</i> port <i>port</i>; } } }</pre> |
| Hierarchy Level | [edit system services extension-service] |
| Release Information | <p>Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.</p> <p>grpc option introduced in Junos OS Release 16.2 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.</p> |
| Description | <p>Allow request-response API execution.</p> <p>Statements are explained separately.</p> |
| Required Privilege Level | <p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p> |
| Related Documentation | <ul style="list-style-type: none"> • Understanding JET Interaction with Junos OS on page 13 • Configuring Request-Response Service for JET Applications on page 24 • extension-service (System Services) on page 32 |

routing-instance (JET Scripts)

| | |
|---|---|
| Syntax | <code>routing-instance <i>routing-instance-name</i>;</code> |
| Hierarchy Level | [edit system extensions extension-service application file filename] |
| Release Information | Statement introduced in Junos OS Release 18.1R1. |
| Description | Configure the routing instance you want to use to update Automation scripts. To use a management instance, configure the management-instance statement along with the routing-instance statement, thus enabling JET scripts to use the non-default management routing instance <code>mgmt_junos</code> when refreshing the scripts. |
| Options | <i>routing-instance-name</i> —Name of the routing instance. For the management instance, use <code>mgmt_junos</code> . Otherwise, you can specify any routing instance name. |
| <div> NOTE: You must also define the routing instance under the [edit routing-instances] hierarchy level.</div> | |
| Required Privilege Level | <code>system</code> —To view this statement in the configuration. <code>system-control</code> —To add this statement to the configuration. |
| Related Documentation | <ul style="list-style-type: none"><i>management-instance</i><i>Management Interface in a Non-Default Instance</i> |

services (System Services)

```

Syntax  services {
        dhcp { # DHCP is not supported on a DCF
            dhcp_services;
        }
        dtcp-only
        flow-tap-dtcp {
            ssh {
                connection-limit limit;
                rate-limit limit;
            }
        }
        extension-service {
            request-response {
                grpc {
                    (clear-text | ssl) {
                        address ip-address;
                        local-certificate local-certificate; # Not specified for cleartext option
                        port port;
                    }
                    max-connections max-connections;
                }
            }
            notification {
                port port;
                max-connections max-connections;
                allow-clients {
                    address ip-address;
                }
            }
            traceoptions {
                file <filename> <files number> <match regex> <size size> <world-readable |
                    no-world-readable>;
                flag flag;
                no-remote-trace;
            }
        }
        finger {
            connection-limit limit;
            rate-limit limit;
        }
        ftp {
            authentication-order [authentication-methods];
            connection-limit limit;
            rate-limit limit;
        }
        service-deployment {
            servers address {
                port-number port-number;
            }
            source-address address;
        }
        ssh {

```

```
authentication-order [authentication-methods];
ciphers [ cipher-1 cipher-2 cipher-3 ...];
client-alive-count-max seconds;
client-alive-interval seconds;
connection-limit limit;
fingerprint-hash (md5 | sha2-256);
hostkey-algorithm (algorithm | no-algorithm);
key-exchange [algorithm];
macs [algorithm];
max-sessions-per-connection <number>;
no-passwords;
no-public-keys;
no-tcp-forwarding;
protocol-version [v1 v2];
rate-limit limit;
root-login (allow | deny | deny-password);
}
resource-monitor {
  free-fw-memory-watermark number;
  free-heap-memory-watermark number;
  free-nh-memory-watermark number;
  high-threshold number;
  no-logging;
  no-throttle;
  resource-category jtree {
    resource-type (contiguous-pages | free-dwords | free-pages) {
      low-watermark number;
      high-watermark number;
    }
  }
}
subscribers-limit {
  client-type (any | dhcp | l2tp | pppoe) {
    chassis {
      limit limit;
    }
    fpc slot-number {
      limit limit;
      pic number {
        limit limit;
        port number {
          limit limit;
        }
      }
    }
  }
}
}
traceoptions {
  file filename <files number> <match regular-expression> <size maximum-file-size>
    <world-readable | no-world-readable>;
  flag flag;
  no-remote-trace;
}
}
subscriber-management {
  enable;
  enforce-strict-scale-limit-license;
```

```

gres-route-flush-delay;
}
overrides {
  interfaces {
    family (inet | inet6) {
      layer2-liveness-detection;
    }
  }
  no-unsolicited-ra;
  ra-initial-interval-max seconds;
  ra-initial-interval-min seconds;
  shmlog {
    disable;
    file filename <files maximum-no-files> <size maximum-file-size>;
    filtering enable;
    log-name {
      all;
      logname {
        <brief | detail | extensive | none | terse>;
        <file-logging | no-file-logging>;
      }
    }
    log-type (debug | info | notice);
  }
}
traceoptions {
  file filename <files number> <match regular-expression> <size maximum-file-size>
    <world-readable | no-world-readable>;
  flag flag;
}
}
telnet {
  authentication-order [authentication-methods];
  connection-limit limit;
  rate-limit limit;
}
web-management {
  http {
    interfaces [ names ];
    port port;
  }
  https {
    interfaces [ names ];
    local-certificate name;
    port port;
  }
  session {
    idle-timeout [ minutes ];
    session-limit [ limit ];
  }
}
xnm-clear-text {
  connection-limit limit;
  rate-limit limit;
}
xnm-ssl {

```

```
    connection-limit limit;  
    local-certificate name;  
    rate-limit limit;  
    ssl-renegotiation;  
  }  
}
```

Hierarchy Level [edit system]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
extension-service option added in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.
grpc option added in Junos OS Release 16.2 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.

Description Configure the router or switch so that users on remote systems can access the local router or switch through the DHCP server, DTCP over SSH, finger, rlogin, SSH, telnet, Web management, Junos XML protocol cleartext, Junos XML protocol SSL, and network utilities, or enable Junos OS to work with the Session and Resource Control (SRC) software. Also, enable configuration of third-party applications developed using the Juniper Extension Toolkit (JET) to run on Junos OS.


The remaining statements are explained separately. Search for a statement in [CLI Explorer](#) or click a linked statement in the Syntax section for details.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *Configuring clear-text or SSL Service for Junos XML Protocol Client Applications*
- *Configuring the Junos OS to Work with SRC Software*
- [Understanding JET Interaction with Junos OS on page 13](#)

source (JET Scripts)

| | |
|---|---|
| Syntax | <code>source url;</code> |
| Hierarchy Level | [edit system extensions extension-service application file <i>filename</i>] |
| Release Information | Statement introduced in Junos OS Release 18.1R1. |
| Description | Specify the location of the master source file for a JET script. When you issue the set refresh configuration mode command at the same hierarchy level, the local copy of the script is overwritten by the version stored at the specified URL. If the load-scripts-from-flash statement is configured, the device refreshes the scripts on the flash drive instead of the hard disk. |
| <div>  NOTE: JET scripts are stored in the <code>/var/db/scripts/jet</code> directory. </div> | |
| Options | url —Master source file for a JET script specified as an HTTP URL, FTP URL, or scp-style remote file specification. |
| Required Privilege Level | maintenance—To view this statement in the configuration. maintenance-control—To add this statement to the configuration. |
| Related Documentation | <ul style="list-style-type: none"> <i>Configuring and Using a Master Source Location for a Script</i> <i>Example: Configuring and Refreshing from the Master Source for a Script</i> |

ssl

Syntax

```
ssl {
    address ip-address;
    local-certificate local-certificate
    mutual-authentication {
        client-certificate-request {
            no-certificate;
            request-certificate;
            request-certificate-and-verify;
            require-certificate;
            require-certificate-and-verify;
        }
    }
    certificate-authority certificate-authority-profile-name;
    port port;
}
```

Hierarchy Level [edit system services extension-service request-response grpc]

Release Information Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.
mutual-authentication, **client-certificate-request**, and **certificate-authority** options introduced in Junos OS Release 17.4R1.

Description Configure API connection settings based on Secure Sockets Layer (SSL) technology.

Options **address *ip-address***—Specify the IP address to listen for incoming connections. If you use the default IP address 0.0.0.0, the JET service process (jsd) listens on the IP address in the default routing instance.

Default: 0.0.0.0

mutual-authentication—Enable bidirectional authentication. Use this option, in conjunction with **client-certificate-request** and **certificate-authority *profile-name*** to configure client authentication using SSL-based certificates.

client-certificate-request—Specify the requirements for a client certificate.

no-certificate—Client certificate is not requested.



NOTE: We strongly recommend that you use this option in a test environment only.

request-certificate—Request certificate from client but do not verify.

request-certificate-and-verify—Request certificate from client and verify if provided.

require-certificate—Client certificate is mandatory, but do not verify.

require-certificate-and-verify—Client certificate is mandatory, and certificate is verified.

Default: no-certificate



NOTE: You can specify only one value for a client certificate.

certificate-authority *profile-name*—Specify the name of a certificate-authority profile configured at the [edit security pki ca-profile] hierarchy level. This profile is used to validate the certificate provided by the client.

port *port*—Specify the port number to accept incoming connections.



NOTE: For gRPC connections used to stream telemetry data, the required port number is 32767.

Range: 1 through 65535

Default: 9090

The remaining statement is explained separately. See [CLI Explorer](#).

| | |
|---------------------------------|--|
| Required Privilege Level | <p>system—To view this statement in the configuration.</p> <p>system-control—To add this statement to the configuration.</p> |
|---------------------------------|--|

| | |
|------------------------------|--|
| Related Documentation | <ul style="list-style-type: none"> • grpc on page 37 • JET Service Process Overview on page 14 • Configuring Request-Response Service for JET Applications on page 24 |
|------------------------------|--|

traceoptions (Extensions)

| | |
|----------------------------|--|
| Syntax | <pre>traceoptions { file <filename> <files number> <match regex> <size size> <world-readable no-world-readable>; flag flag; no-remote-trace; }</pre> |
| Hierarchy Level | [edit system extensions extension-service application] |
| Release Information | Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | Trace options for extension-service applications. |
| Options | <p>file—Indicate trace file information.</p> <p>filename—Name of the file to receive the tracing operation output. Enclose the name in quotation marks. Traceoption output files are located in the <code>/var/log/</code> directory.</p> <p>files number—(Optional) Specify maximum number of trace files. Range: 2 through 1000 Default: 3</p> <p>size size—(Optional) Specify the maximum size of each trace file. When a trace file named trace-file reaches its maximum size, it is renamed trace-file.0. The traceoption output continues in a second trace file named trace-file.1. When trace-file.1 reaches its maximum size, output continues in a third file named trace-file.2, and so on. When the maximum number of trace files is reached, the oldest trace file is overwritten. Range: 10240 through 1073741824 Default: 128k</p> <p>world-readable no-world-readable—(Optional). Grant all users permission to read log files, or restrict the permission only to the root user and users who have Junos OS maintenance permission.</p> <p>flag flag—Specify the tracing operation to perform. To specify more than one tracing operation, include multiple flag statements:</p> <p>all—Trace all operations.</p> <p>config—Trace important events.</p> <p>general—Trace script input data.</p> <p>grpc—Trace grpc server events.</p> |

notification—Trace notification events.

routing-socket—Trace routing socket calls.

timeouts—Trace timeouts.

timer—Trace internal timer events.

no-remote-trace—Disable remote tracing. This option is valid only when [system tracing] is configured.

| | |
|---------------------------------|---|
| Required Privilege Level | trace—To view this statement in the configuration. trace-control—To add this statement to the configuration. |
|---------------------------------|---|

| | |
|------------------------------|--|
| Related Documentation | • application on page 29 |
|------------------------------|--|

traceoptions (Services)

| | |
|---------------------|---|
| Syntax | <pre>traceoptions { file <filename> <files number> <match regex> <size size> <world-readable no-world-readable>; flag flag; no-remote-trace; }</pre> |
| Hierarchy Level | [edit system services extension-service] |
| Release Information | Statement introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | Define tracing operations for the JET service process (jsd). |
| Options | <p>file—Indicate trace file information.</p> <p>filename—Name of the file to receive the tracing operation output. Enclose the name in quotation marks. Traceoption output files are located in the /var/log/ directory.</p> <p>files number—(Optional) Specify the maximum number of trace files. Range: 2 through 1000 Default: 10</p> <p>match regex—Specify the regular expression for lines to be logged.</p> <p>size size—(Optional) Specify the maximum size of each trace file. When a trace file named trace-file reaches its maximum size, it is renamed trace-file.0. The traceoption output continues in a second trace file named trace-file.1. When trace-file.1 reaches its maximum size, output continues in a third file named trace-file.2, and so on. When the maximum number of trace files is reached, the oldest trace file is overwritten. Range: 10,240 through 1,073,741,824 bytes Default: 1000k</p> <p>world-readable no-world-readable—(Optional). Grant all users permission to read log files, or restrict the permission only to the root user and users who have Junos OS maintenance permission.</p> <p>flag flag—Specify the tracing operation to perform. To specify more than one tracing operation, include multiple flag statements:</p> <ul style="list-style-type: none">• all—Trace everything.• config—Trace configuration events.• general—Trace general events.• notification—Trace notification events. |

- **routing-socket**—Trace routing socket calls
- **grpc**—Trace grpc server events.
- **timeouts**—Trace timeouts.
- **timer**—Trace internal timer events.

no-remote-trace—Disable remote tracing.

Required Privilege Level **trace**—To view this statement in the configuration.
 trace-control—To add this statement to the configuration.

Related Documentation • *extension-service*

traceoptions (routing-options programmable-rpd)

| | |
|---------------------|--|
| Syntax | <pre>traceoptions { file <i>filename</i> <files <i>number</i>> <size <i>size</i>> <world-readable no-world-readable>; flag <i>flag</i> <disable>; }</pre> |
| Hierarchy Level | <pre>[edit routing-options programmable-rpd] flag <<i>flags</i>> file <<i>filename</i>> <<i>size</i>></pre> |
| Release Information | Statement introduced in Junos OS Release 16.2 for MX Series. |
| Description | <p>Starts logging traces related to the programmable routing protocol process (prpd). The prpd provides public APIs to program routing systems, making it possible for users to directly access the APIs to customize, create and modify behavior of their network.</p> <p>Use the traceoptions command, along with related show commands, to help debug client-server interactions, identify the flow of control, and detect errors, get client-level information and statistics.</p> <p>You can filter traces according to the flag(s) you have enabled.</p> |
| Default | If you do not include this statement, no tracing operations are performed. |
| Options | <p>Values:</p> <p>file <i>filename</i>—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory /var/log.</p> <p>files <i>number</i>—(Optional) Maximum number of trace files. When a trace file named trace-file reaches its maximum size, it is renamed trace-file.0, then trace-file.1, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten. Note that if you specify a maximum number of files, you also must specify a maximum file size with the size option.</p> <p>Range: 2 through 1000 files</p> <p>Default: 10 files</p> <p>no-world-readable—(Optional) Prevent any user from reading the log file.</p> <p>size <i>size</i>—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named trace-file reaches this size, it is renamed trace-file.0. When the trace-file again reaches its maximum size, trace-file.0 is renamed trace-file.1 and trace-file is renamed trace-file.0. This renaming scheme continues until the maximum number of trace files is reached. Then, the oldest trace file is overwritten. Note that if you specify a maximum file size, you also must specify a maximum number of trace files with the files option.</p> <p>Syntax: xk to specify KB, xm to specify MB, or xg to specify GB</p> |

Range: 1024 to 4,294,967,295 bytes

Default: 128 KB

world-readable—(Optional) Allow any user to read the log file.

flag *flag*—Specifies the tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. The options are explained here:

- **all**—All tracing operations
- **client**—Client events
- **general**—All normal operations and routing table changes (a combination of the **normal** and **route** trace operations)
- **normal**—All normal operations
- **policy**—Routing policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Interface transactions and processing
- **timer**—Timer usage

| | |
|---------------------------------|---|
| Required Privilege Level | routing and trace—To view this statement in the configuration. |
| | routing-control and trace-control—To add this statement to the configuration. |

| | |
|------------------------------|--|
| Related Documentation | • purge-timeout on page 44 |
| | • show programmable-rpd clients on page 67 |
| | • show route on page 68 |

CHAPTER 5

Operational Commands

- request extension-service (start | stop)
- show extension-service status
- show programmable-rpd clients
- show route
- show route detail

request extension-service (start | stop)

| | |
|---------------------------------|---|
| Syntax | <code>request extension-service (start stop) <i>application-name</i></code> <code><invoke-debugger cli></code> |
| Release Information | Command introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series. |
| Description | Start or stop a JET application running on a device running Junos OS. |
| Options | <i>application-name</i> —Name of application to be started or stopped. <i>invoke-debugger cli</i> —(Optional) Starts the extension service process in debugger mode. |
| Required Privilege Level | maintenance |
| Related Documentation | <ul style="list-style-type: none">• Starting and Stopping a JET Application Running on a Device Running Junos OS on page 19 |
| Output Fields | When you enter this command, you are provided feedback on the status of your request. |

Sample Output

```
user@device> request extension-service start cmdline_args.py
Extension-service application 'cmdline_args.py' started with pid: 99418
```


show extension-service status

Syntax `show extension-service status (application-name | all)`

Release Information Command introduced in Junos OS Release 16.1 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.

Description Display the status of all JET applications.



NOTE: The `show extension-service status` operational command is limited to use with Python applications only.

Options *application-name*—Display information for a single application.

all—Display information for all JET applications running on the system.

Required Privilege Level view

Related Documentation

- [Starting and Stopping a JET Application Running on a Device Running Junos OS on page 19](#)

List of Sample Output

- [show extension-service status on page 66](#)
- [show extension-service status all on page 66](#)
- [show extension-service status all \(when no applications are active\) on page 66](#)

Output Fields [Table 3 on page 65](#) lists the output fields for the `show extension-service status` command.

Table 3: show extension-service status Output Fields

| Field Name | Field Description |
|--------------------|--------------------------------------|
| Name | Name of the application. |
| Arguments | Arguments passed to the application. |
| Process-id | Process ID. |
| Stack-Segment-Size | Size of the stack segment memory. |
| Data-Segment-Size | Size of the data segment memory. |

Sample Output

show extension-service status

```
user@host> show extension-service status application-one
Extension service application details:
Name : application-one
Arguments: -arg1 foo -arg2 goo
Process-id: 52592
Stack-Segment-Size: 16777216B
Data-Segment-Size: 134217728B
```

show extension-service status all

```
user@host> show extension-service status all
Extension service application details:
Name : application-name1
Arguments: -arg1 foo -arg2 goo
Process-id: 54834
Stack-Segment-Size: 16777216B
Data-Segment-Size: 134217728B
Name : application-name2
Arguments: -arg1 foo -arg2 goo
Process-id: 55011
Stack-Segment-Size: 16777216B
Data-Segment-Size: 134217728B
```

show extension-service status all (when no applications are active)

```
user@host> show extension-service status all
warning: No active extension-services
```

show programmable-rpd clients

Syntax show programmable-rpd clients

Release Information Command introduced in Junos OS Release 16.2 for MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series.

Description Lists clients connected to the programmable routing protocol process (prpd) server. The prpd provides public APIs to program routing systems, making it possible for users to directly access the APIs to customize, create and modify behavior of their network.

Output provided with the command includes client specific details and statistics such as client ID, protocol and corresponding gateway handle, purge timer, the client up/down status, and if the client is disconnected, the time remaining before the client state is purged (if the client has not registered any protocol, the gateway handle is 0).

Required Privilege Level view

Related Documentation

- [purge-timeout on page 44](#)
- [show programmable-rpd clients on page 67](#)
- [show route on page 68](#)

List of Sample Output [show programmable-rpd clients on page 67](#)

Output Fields

Sample Output

show programmable-rpd clients

```
user@host> show programmable-rpd clients
RPD global purge timeout: 120
RPD Server connected client details:
ClientIdentifier  PurgeTimer  Status  Timeout  Protocol  Gateway
3                150         Up      117      BGP-Static  578
2                75          Up      117      NoGwProtocol 0
1                120         Down    117      BGP-Static  577
```

show route

List of Syntax [Syntax on page 68](#)
 [Syntax \(EX Series Switches\) on page 68](#)

Syntax show route
 <all>
 <destination-prefix>
 <logical-system (all | *logical-system-name*)>
 <private>
 <te-ipv4-prefix-ip *te-ipv4-prefix-ip*>
 <te-ipv4-prefix-node-ip *te-ipv4-prefix-node-ip*>
 <te-ipv4-prefix-node-iso *te-ipv4-prefix-node-iso*>

Syntax (EX Series Switches) show route
 <all>
 <destination-prefix>
 <private>

Release Information Command introduced before Junos OS Release 7.4.
 Command introduced in Junos OS Release 9.0 for EX Series switches.
 Option **private** introduced in Junos OS Release 9.5.
 Option **private** introduced in Junos OS Release 9.5 for EX Series switches.
 Command introduced in Junos OS Release 15.1R3 on MX Series routers for enhanced subscriber management.
 Option **display-client-data** introduced in Junos OS Release 16.2R1 on MX80, MX104, MX240, MX480, MX960, MX2010, MX2020, vMX Series routers.
 Options **te-ipv4-prefix-ip**, **te-ipv4-prefix-node-ip**, and **te-ipv4-prefix-node-iso** introduced in Junos OS Release 17.2R1 on MX Series and PTX Series.

Description Display the active entries in the routing tables.

Options **none**—Display brief information about all active entries in the routing tables.

all—(Optional) Display information about all routing tables, including private, or internal, routing tables.

destination-prefix—(Optional) Display active entries for the specified address or range of addresses.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

private—(Optional) Display information only about all private, or internal, routing tables.

display-client-data —(Optional) Display client id and cookie information for routes installed by the routing protocol process client applications.

te-ipv4-prefix-ip *te-ipv4-prefix-ip*—(Optional) Display IPv4 address of the traffic-engineering prefix, without the mask length if present in the routing table.

te-ipv4-prefix-node-ip *te-ipv4-prefix-node-ip*—(Optional) Display all prefixes that have originated from the traffic-engineering node. You can filter IPv4 node addresses from the traffic-engineered routes in the **lsdist.0** table.

te-ipv4-prefix-node-iso *te-ipv4-prefix-node-iso*—(Optional) Display all prefixes that have originated from the traffic-engineering node. You can filter IPv4 routes with the specified ISO circuit ID from the **lsdist.0** table.

Required Privilege Level

view

Related Documentation

- *Understanding IS-IS Configuration*
- *Example: Configuring IS-IS*
- *Examples: Configuring Internal BGP Peering*
- *Examples: Configuring External BGP Peering*
- *Examples: Configuring OSPF Routing Policy*
- *Verifying and Managing Junos OS Enhanced Subscriber Management*

List of Sample Output

[show route on page 72](#)
[show route \(VPN\) on page 73](#)
[show route \(with Destination Prefix\) on page 73](#)
[show route destination-prefix detail on page 73](#)
[show route extensive on page 73](#)
[show route extensive \(ECMP\) on page 74](#)
[show route extensive \(Multipath Resolution\) on page 74](#)
[show route \(Enhanced Subscriber Management\) on page 78](#)
[show route \(IPv6 Flow Specification\) on page 79](#)
[show route display-client-data detail on page 79](#)
[show route te-ipv4-prefix-ip on page 80](#)
[show route te-ipv4-prefix-ip extensive on page 80](#)
[show route te-ipv4-prefix-node-iso on page 83](#)
[show route te-ipv4-prefix-node-iso extensive on page 83](#)
[show route te-ipv4-prefix-node-iso detail on page 86](#)

Output Fields

[Table 4 on page 69](#) describes the output fields for the **show route** command. Output fields are listed in the approximate order in which they appear.

Table 4: show route Output Fields

| Field Name | Field Description |
|----------------------------|---|
| <i>routing-table-name</i> | Name of the routing table (for example, inet.0). |
| <i>number destinations</i> | Number of destinations for which there are routes in the routing table. |

Table 4: show route Output Fields (continued)

| Field Name | Field Description |
|---|--|
| <i>number routes</i> | <p>Number of routes in the routing table and total number of routes in the following states:</p> <ul style="list-style-type: none"> • active (routes that are active). • holddown (routes that are in the pending state before being declared inactive). A holddown route was once the active route and is no longer the active route. The route is in the holddown state because a protocol still has interest in the route, meaning that the interest bit is set. A protocol might have its interest bit set on the previously active route because the protocol is still advertising the route. The route will be deleted after all protocols withdraw their advertisement of the route and remove their interest bit. A persistent holddown state often means that the interested protocol is not releasing its interest bit properly. <p>However, if you have configured advertisement of multiple routes (with the add-path or advertise-inactive statement), the holddown bit is most likely set because BGP is advertising the route as an active route. In this case, you can ignore the holddown state because nothing is wrong.</p> <ul style="list-style-type: none"> • hidden (routes that are not used because of a routing policy). |
| <i>destination-prefix</i> | <p>Route destination (for example:10.0.0.1/24). Sometimes the route information is presented in another format, such as:</p> <ul style="list-style-type: none"> • MPLS-label (for example, 80001). • interface-name (for example, ge-1/0/2). • neighbor-address:control-word-status:encapsulation type:vc-id:source (Layer 2 circuit only. For example, 10.1.1.195:NoCtrlWord:1:1:Local/96): <ul style="list-style-type: none"> • neighbor-address—Address of the neighbor. • control-word-status—Whether the use of the control word has been negotiated for this virtual circuit: NoCtrlWord or CtrlWord. • encapsulation type—Type of encapsulation, represented by a number: (1) Frame Relay DLCI, (2) ATM AAL5 VCC transport, (3) ATM transparent cell transport, (4) Ethernet, (5) VLAN Ethernet, (6) HDLC, (7) PPP, (8) ATM VCC cell transport, (10) ATM VPC cell transport. • vc-id—Virtual circuit identifier. • source—Source of the advertisement: Local or Remote. |
| <i>[protocol, preference]</i> | <p>Protocol from which the route was learned and the preference value for the route.</p> <ul style="list-style-type: none"> • +—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • -—A hyphen indicates the last active route. • *—An asterisk indicates that the route is both the active and the last active route. An asterisk before a to line indicates the best subpath to the route. <p>In every routing metric except for the BGP LocalPref attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the LocalPref value in the Preference2 field. For example, if the LocalPref value for Route 1 is 100, the Preference2 value is -101. If the LocalPref value for Route 2 is 155, the Preference2 value is -156. Route 2 is preferred because it has a higher LocalPref value and a lower Preference2 value.</p> |
| <i>weeks:days hours:minutes:seconds</i> | How long the route been known (for example, 2w4d 13:11:14 , or 2 weeks, 4 days, 13 hours, 11 minutes, and 14 seconds). |
| <i>metric</i> | Cost value of the indicated route. For routes within an AS, the cost is determined by the IGP and the individual protocol metrics. For external routes, destinations, or routing domains, the cost is determined by a preference value. |

Table 4: show route Output Fields (continued)

| Field Name | Field Description |
|-------------------------|---|
| localpref | Local preference value included in the route. |
| from | Interface from which the route was received. |
| AS path | <p>AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated:</p> <ul style="list-style-type: none"> • I—IGP. • E—EGP. • ?—Incomplete; typically, the AS path was aggregated. <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> • []—Brackets enclose the local AS number associated with the AS path if more than one AS number is configured on the routing device, or if AS path prepending is configured. • { }—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • ()—Parentheses enclose a confederation. • ([])—Parentheses and brackets enclose a confederation set. <p>NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.</p> |
| encapsulated | Extended next-hop encoding capability enabled for the specified BGP community for routing IPv4 traffic over IPv6 tunnels. When BGP receives routes without the tunnel community, IPv4-Over IPv6 tunnels are not created and BGP routes are resolved without encapsulation. |
| Route Labels | Stack of labels carried in the BGP route update. |
| validation-state | <p>(BGP-learned routes) Validation status of the route:</p> <ul style="list-style-type: none"> • Invalid—Indicates that the prefix is found, but either the corresponding AS received from the EBGp peer is not the AS that appears in the database, or the prefix length in the BGP update message is longer than the maximum length permitted in the database. • Unknown—Indicates that the prefix is not among the prefixes or prefix ranges in the database. • Unverified—Indicates that the origin of the prefix is not verified against the database. This is because the database got populated and the validation is not called for in the BGP import policy, although origin validation is enabled, or the origin validation is not enabled for the BGP peers. • Valid—Indicates that the prefix and autonomous system pair are found in the database. |
| to | <p>Next hop to the destination. An angle bracket (>) indicates that the route is the selected route.</p> <p>If the destination is Discard, traffic is dropped.</p> |

Table 4: show route Output Fields (continued)

| Field Name | Field Description |
|-----------------|--|
| via | <p>Interface used to reach the next hop. If there is more than one interface available to the next hop, the interface that is actually used is followed by the word Selected. This field can also contain the following information:</p> <ul style="list-style-type: none"> • Weight—Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible. • Balance—Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a routing device is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing. • lsp-path-name—Name of the LSP used to reach the next hop. • label-action—MPLS label and operation occurring at the next hop. The operation can be pop (where a label is removed from the top of the stack), push (where another label is added to the label stack), or swap (where a label is replaced by another label). For VPNs, expect to see multiple push operations, corresponding to the inner and outer labels required for VPN routes (in the case of a direct PE-to-PE connection, the VPN route would have the inner label push only). |
| Private unicast | (Enhanced subscriber management for MX Series routers) Indicates that an access-internal route is managed by enhanced subscriber management. By contrast, access-internal routes <i>not</i> managed by enhanced subscriber management are displayed with associated next-hop and media access control (MAC) address information. |
| balance | Distribution of the load based on the underlying operational interface bandwidth for equal-cost multipaths (ECMP) across the nexthop gateways in percentages. |

Sample Output

show route

```

user@host> show route
inet.0: 11 destinations, 12 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:65500:1:10.0.0.20/240
    * [MVPN/70] 19:53:41, metric2 1
    Indirect
1:65500:1:10.0.0.40/240
    * [BGP/170] 19:53:29, localpref 100, from 10.0.0.30
    AS path: I
    > to 10.0.24.4 via lt-0/3/0.24, label-switched-path toD
    [BGP/170] 19:53:26, localpref 100, from 10.0.0.33
    AS path: I
    > to 10.0.24.4 via lt-0/3/0.24, label-switched-path toD
1:65500:1:10.0.0.60/240
    * [BGP/170] 19:53:29, localpref 100, from 10.0.0.30
    AS path: I
    > to 10.0.28.8 via lt-0/3/0.28, label-switched-path toF
    [BGP/170] 19:53:25, localpref 100, from 10.0.0.33
    AS path: I
    > to 10.0.28.8 via lt-0/3/0.28, label-switched-path toF

```


show route (VPN)

The following sample output shows a VPN route with composite next hops enabled. The first **Push** operation corresponds to the outer label. The second **Push** operation corresponds to the inner label.

```
user@host> show route 192.0.2.0

13979:665001.inet.0: 871 destinations, 3556 routes (871 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

192.0.2.0/24          [BGP/170] 00:28:32, localpref 100, from 10.9.9.160
                     AS path: 13980 ?, validation-state: unverified
                     > to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)
                     [BGP/170] 00:28:28, localpref 100, from 10.9.9.169
                     AS path: 13980 ?, validation-state: unverified
                     > to 10.100.0.42 via ae2.0, Push 126016, Push 300368(top)
                     #[Multipath/255] 00:28:28, metric2 102
                     > to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)
                     to 10.100.0.42 via ae2.0, Push 16, Push 300368(top)
```

show route (with Destination Prefix)

```
user@host> show route 192.168.0.0/12

inet.0: 10 destinations, 10 routes (9 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.0/12      *[Static/5] 2w4d 12:54:27
                     > to 192.168.167.254 via fxp0.0
```

show route destination-prefix detail

```
user@host> show route 198.51.100.0 detail

inet.0: 15 destinations, 20 routes (15 active, 0 holddown, 0 hidden)
198.51.100.0/24 (2 entries, 2 announced)
  *BGP      Preference: 170/-101
  ...
  BGP-Static Preference: 4294967292
    Next hop type: Discard
    Address: 0x9041ae4
    Next-hop reference count: 2
    State: <NoReadvrt Int Ext AlwaysFlash>
  Inactive reason: Route Preference
  Local AS: 200
  Age: 4d 1:40:40
  Validation State: unverified
  Task: RT
  Announcement bits (1): 2-BGP_RT_Background
  AS path: 4 5 6 I
```

show route extensive

```
user@host> show route extensive

v1.mvpn.0: 5 destinations, 8 routes (5 active, 1 holddown, 0 hidden)
1:65500:1:10.0.0.40/240 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
```

```

PMSI: Flags 0x0: Label[0:0:0]: PIM-SM: Sender 10.0.0.40 Group
203.0.113.1
Next hop type: Indirect
Address: 0x92455b8
Next-hop reference count: 2
Source: 10.0.0.30
Protocol next hop: 10.0.0.40
Indirect next hop: 2 no-forward
State: <Active Int Ext>
    Local AS: 64510 Peer AS: 64511
Age: 3 Metric2: 1
Validation State: unverified
Task: BGP_64510.10.0.0.30+179
Announcement bits (2): 0-PIM.v1 1-mvpn global task
AS path: I (Originator) Cluster list: 10.0.0.30
AS path: Originator ID: 10.0.0.40
Communities: target:64502:100 encapsulation:0L:14 Import
Accepted
Localpref: 100
Router ID: 10.0.0.30
Primary Routing Table bgp.mvpn.0
Indirect next hops: 1
    Protocol next hop: 10.0.0.40 Metric: 1
    Indirect next hop: 2 no-forward
    Indirect path forwarding next hops: 1
        Next hop type: Router
        Next hop: 10.0.24.4 via lt-0/3/0.24 weight 0x1
    10.0.0.40/32 Originating RIB: inet.3
        Metric: 1 Node path count: 1
        Forwarding nexthops: 1
            Nexthop: 10.0.24.4 via lt-0/3/0.24

```

show route extensive (ECMP)

```

user@host> show route extensive
*IS-IS Preference: 15
    Level: 1
    Next hop type: Router, Next hop index: 1048577
    Address: 0xFFFFFFFF
    Next-hop reference count: YY
    Next hop: 198.51.100.2 via ae1.0 balance 43%, selected
    Session Id: 0x141
    Next hop: 192.0.2.2 via ae0.0 balance 57%

```

show route extensive (Multipath Resolution)

```

user@host> show route extensive
inet.0: 37 destinations, 37 routes (36 active, 0 holddown, 1 hidden)
10.1.1.2/32 (1 entry, 1 announced)
TSI:
KRT in-kernel 10.1.1.2/32 -> {indirect(1048574)}
*Static Preference: 5
    Next hop type: Indirect, Next hop index: 0
    Address: 0xb39d1b0
    Next-hop reference count: 2
    Next hop type: Router, Next hop index: 581
    Next hop: 10.1.1.2 via ge-2/0/1.0, selected
    Session Id: 0x144
    Next hop: 10.2.1.2 via ge-2/0/2.0, selected
    Session Id: 0x145

```

```

Protocol next hop: 10.1.1.1
Indirect next hop: 0xb2b20f0 1048574 INH Session ID: 0x143
State: <Active Int Ext>
Age: 2:53 Metric2: 0
Validation State: unverified
Task: RT
Announcement bits (2): 0-KRT 2-Resolve tree 1
AS path: I
Indirect next hops: 1
    Protocol next hop: 10.1.1.1
    Indirect next hop: 0xb2b20f0 1048574 INH Session ID: 0x143

    Indirect path forwarding next hops: 2
        Next hop type: Router
        Next hop: 10.1.1.2 via ge-2/0/1.0
        Session Id: 0x144
        Next hop: 10.2.1.2 via ge-2/0/2.0
        Session Id: 0x145
10.1.1.1/32 Originating RIB: inet.0
    Node path count: 1
    Node flags: 1
    Forwarding nexthops: 2 (Merged)
    Nexthop: 10.1.1.2 via ge-2/0/1.0
    Nexthop: 10.2.1.2 via ge-2/0/2.0

user@host> show route active-path extensive
user@host> show route 198.51.100.1 active-path extensive

inet.0: 1000061 destinations, 1000082 routes (1000061 active, 0 holddown, 0 hidden)
198.51.100.1/32 (1 entry, 1 announced)
TSI:
KRT in-kernel 198.51.100.1/32 -> {indirect(1051215)}
unicast reverse-path: 0
[ae0.0 ae1.0]
Page 0 idx 0, (group Internet-IPv4 type External) Type 1 val 0xbb2e53d8 (adv_entry)
Advertised metrics:
Nexthop: Self
AS path: [500] 410 I
Communities:
Path 198.51.100.1 from 10.0.0.11 Vector len 4. Val: 0
*BGP Preference: 170/-101
Next hop type: Indirect, Next hop index: 0
Address: 0x2e9aacdc
Next-hop reference count: 500000
Source: 10.0.0.11
Next hop type: Router, Next hop index: 0
Next hop: 10.0.12.2 via ae0.0 weight 0x1
Label operation: Push 3851, Push 25, Push 20(top)
Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
Load balance label: Label 3851: None; Label 25: None; Label 20: None;
Label element ptr: 0xb5dc1780
Label parent element ptr: 0x18d48080
Label element references: 2
Label element child references: 0
Label element lsp id: 0
Session Id: 0x0
Next hop: 10.0.12.2 via ae0.0 weight 0x1
Label operation: Push 3851, Push 25, Push 22(top)
Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
Load balance label: Label 3851: None; Label 25: None; Label 22: None;
Label element ptr: 0xb5dc1700
Label parent element ptr: 0x18d41000

```

Label element references: 2
Label element child references: 0
Label element lsp id: 0
Session Id: 0x0
Next hop: 10.0.12.2 via ae0.0 weight 0x1
Label operation: Push 3851, Push 24, Push 48(top)
Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
Load balance label: Label 3851: None; Label 24: None; Label 48: None;
Label element ptr: 0x18d40800
Label parent element ptr: 0x18d49780
Label element references: 3
Label element child references: 0
Label element lsp id: 0
Session Id: 0x0
Next hop: 10.0.12.2 via ae0.0 weight 0x1
Label operation: Push 3851, Push 24, Push 49(top)
Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
Load balance label: Label 3851: None; Label 24: None; Label 49: None;
Label element ptr: 0xb5dc1680
Label parent element ptr: 0x18d48f00
Label element references: 2
Label element child references: 0
Label element lsp id: 0
Session Id: 0x0
Next hop: 10.0.13.3 via ae1.0 weight 0x1
Label operation: Push 3851, Push 25, Push 21(top)
Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
Load balance label: Label 3851: None; Label 25: None; Label 21: None;
Label element ptr: 0xb5dc1600
Label parent element ptr: 0x18d44d80
Label element references: 2
Label element child references: 0
Label element lsp id: 0
Session Id: 0x0
Next hop: 10.0.13.3 via ae1.0 weight 0x1
Label operation: Push 3851, Push 25, Push 25(top)
Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
Load balance label: Label 3851: None; Label 25: None; Label 25: None;
Label element ptr: 0xb5dc1580
Label parent element ptr: 0x18d3da80
Label element references: 2
Label element child references: 0
Label element lsp id: 0
Session Id: 0x0
Next hop: 10.0.13.3 via ae1.0 weight 0x1, selected
Label operation: Push 3851, Push 24, Push 68(top)
Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
Load balance label: Label 3851: None; Label 24: None; Label 68: None;
Label element ptr: 0x18d41500
Label parent element ptr: 0x18d49000
Label element references: 3
Label element child references: 0
Label element lsp id: 0
Session Id: 0x0
Next hop: 10.0.13.3 via ae1.0 weight 0x1
Label operation: Push 3851, Push 24, Push 69(top)
Label TTL action: prop-ttl, prop-ttl, prop-ttl(top)
Load balance label: Label 3851: None; Label 24: None; Label 69: None;
Label element ptr: 0xb5dc1500
Label parent element ptr: 0x18d48300
Label element references: 2

```

Label element child references: 0
Label element lsp id: 0
Session Id: 0x0
Protocol next hop: 10.0.0.11
Label operation: Push 3851
Label TTL action: prop-ttl
Load balance label: Label 3851: None;
Indirect next hop: 0x1883e200 1051215 INH Session ID: 0xb0d
State:
Local AS: 500 Peer AS: 500
Age: 1:40:03 Metric2: 2
Validation State: unverified
Task: BGP_500.10.0.0.11
Announcement bits (5): 0-KRT 8-KRT 9-BGP_RT_Background 10-Resolve tree 5 11-Resolve
tree 8
AS path: 410 I
Accepted
Route Label: 3851
Localpref: 100
Router ID: 10.0.0.11
Indirect next hops: 1
Protocol next hop: 10.0.0.11 Metric: 2
Label operation: Push 3851
Label TTL action: prop-ttl
Load balance label: Label 3851: None;
Indirect next hop: 0x1883e200 1051215 INH Session ID: 0xb0d
Indirect path forwarding next hops (Merged): 8
Next hop type: Router
Next hop: 10.0.12.2 via ae0.0 weight 0x1
Session Id: 0x0
Next hop: 10.0.12.2 via ae0.0 weight 0x1
Session Id: 0x0
Next hop: 10.0.12.2 via ae0.0 weight 0x1
Session Id: 0x0
Next hop: 10.0.12.2 via ae0.0 weight 0x1
Session Id: 0x0
Next hop: 10.0.13.3 via ae1.0 weight 0x1
Session Id: 0x0
Next hop: 10.0.13.3 via ae1.0 weight 0x1
Session Id: 0x0
Next hop: 10.0.13.3 via ae1.0 weight 0x1
Session Id: 0x0
Next hop: 10.0.13.3 via ae1.0 weight 0x1
Session Id: 0x0
10.0.0.11/32 Originating RIB: inet.3
Metric: 1 Node path count: 4
Node flags: 1
Indirect nexthops: 4
Protocol Nexthop: 10.0.0.4 Metric: 1 Push 24
Indirect nexthop: 0x1880f200 1048597 INH Session ID: 0xb0c
Path forwarding nexthops link: 0x36120400
Path inh link: 0x0
Indirect path forwarding nexthops: 2
Nexthop: 10.0.12.2 via ae0.0
Session Id: 0
Nexthop: 10.0.13.3 via ae1.0
Session Id: 0
10.0.0.4/32 Originating RIB: inet.3
Metric: 1 Node path count: 1
Forwarding nexthops: 2
Nexthop: 10.0.12.2 via ae0.0

```

```
Session Id: 0
Nexthop: 10.0.13.3 via ae1.0
Session Id: 0
Protocol Nexthop: 10.0.0.5 Metric: 1 Push 24
Indirect nexthop: 0x18810000 1048596 INH Session ID: 0xb0b
Path forwarding nexthops link: 0x1545be00
Path inh link: 0x0
Indirect path forwarding nexthops: 2
Nexthop: 10.0.12.2 via ae0.0
Session Id: 0
Nexthop: 10.0.13.3 via ae1.0
Session Id: 0
10.0.0.5/32 Originating RIB: inet.3
Metric: 1 Node path count: 1
Forwarding nexthops: 2
Nexthop: 10.0.12.2 via ae0.0
Session Id: 0
Nexthop: 10.0.13.3 via ae1.0
Session Id: 0
Protocol Nexthop: 10.0.0.6 Metric: 1 Push 25
Indirect nexthop: 0x1880e600 1048588 INH Session ID: 0xb0a
Path forwarding nexthops link: 0x3611f440
Path inh link: 0x0
Indirect path forwarding nexthops: 2
Nexthop: 10.0.12.2 via ae0.0
Session Id: 0
Nexthop: 10.0.13.3 via ae1.0
Session Id: 0
10.0.0.6/32 Originating RIB: inet.3
Metric: 1 Node path count: 1
Forwarding nexthops: 2
Nexthop: 10.0.12.2 via ae0.0
Session Id: 0
Nexthop: 10.0.13.3 via ae1.0
Session Id: 0
Protocol Nexthop: 10.0.0.7 Metric: 1 Push 25
Indirect nexthop: 0x1880dc00 1048586 INH Session ID: 0xb09
Path forwarding nexthops link: 0x15466d80
Path inh link: 0x0
Indirect path forwarding nexthops: 2
Nexthop: 10.0.12.2 via ae0.0
Session Id: 0
Nexthop: 10.0.13.3 via ae1.0
Session Id: 0
10.0.0.7/32 Originating RIB: inet.3
Metric: 1 Node path count: 1
Forwarding nexthops: 2
Nexthop: 10.0.12.2 via ae0.0
Session Id: 0
Nexthop: 10.0.13.3 via ae1.0
Session Id: 0
```

show route (Enhanced Subscriber Management)

```
user@host> show route
inet.0: 41 destinations, 41 routes (40 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

198.51.100.11/24    *[Access-internal/12] 00:00:08
> to #0 10.0.0.1.93.65 via demux0.1073741824
```

```
198.51.100.12/24    *[Access-internal/12] 00:00:08
                  Private unicast
```

show route (IPv6 Flow Specification)

```
user@host> show route
inet6.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8::10:255:185:19/128
    *[Direct/0] 05:11:27
    > via lo0.0
2001:db8::11:11:11:0/120
    *[BGP/170] 00:28:58, localpref 100
    AS path: 2000 I, validation-state: unverified
    > to 2001:db8::13:14:2:2 via ge-1/1/4.0
2001:db8::13:14:2:0/120*[Direct/0] 00:45:07
    > via ge-1/1/4.0
2001:db8::13:14:2:1/128*[Local/0] 00:45:18
    Local via ge-1/1/4.0
fe80::2a0:a50f:fc71:71d5/128
    *[Direct/0] 05:11:27
    > via lo0.0
fe80::5e5e:abff:feb0:933e/128
    *[Local/0] 00:45:18
    Local via ge-1/1/4.0

inet6flow.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8::11:11:11:10/128,*,proto=6,dstport=80,srcport=65535/term:1
    *[BGP/170] 00:28:58, localpref 100, from 2001:db8::13:14:2:2
    AS path: 2000 I, validation-state: unverified
    Fictitious
2001:db8::11:11:11:30/128,*,icmp6-type=128,len=100,dscp=10/term:2
    *[BGP/170] 00:20:54, localpref 100, from 2001:db8::13:14:2:2
    AS path: 2000 I, validation-state: unverified
    Fictitious
```

show route display-client-data detail

```
user@host> show route 198.51.100.0/24 display-client-data detail
inet.0: 59 destinations, 70 routes (59 active, 0 holddown, 0 hidden)
198.51.100.0/24 (1 entry, 1 announced)
    State: <FlashAll>
    *BGP-Static Preference: 5/-101
    Next hop type: Indirect, Next hop index: 0
    Address: 0xa5c2af8
    Next-hop reference count: 2
    Next hop type: Router, Next hop index: 1641
    Next hop: 192.0.2.1 via ge-2/1/1.0, selected
    Session Id: 0x160
    Protocol next hop: 192.0.2.1
    Indirect next hop: 0xa732cb0 1048621 INH Session ID: 0x17e
    State: <Active Int Ext AlwaysFlash NSR-incapable Programmed>
    Age: 3:13 Metric2: 0
    Validation State: unverified
    Announcement bits (3): 0-KRT 5-LDP 6-Resolve tree 3
```

```
AS path: I
Client id: 1, Cookie: 1
```

show route te-ipv4-prefix-ip

```
user@host> show route te-ipv4-prefix-ip 10.10.10.10
lsdist.0: 283 destinations, 283 routes (283 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L1:0
}/1152
      *[IS-IS/15] 00:01:01
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0101.0101.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:01:01
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0202.0202.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:01:01
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0303.0303.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:01:01
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0404.0404.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:01:01
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0505.0505.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:01:01
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0606.0606.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:01:01
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0707.0707.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:01:01
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:01:01
      Fictitious
```

show route te-ipv4-prefix-ip extensive

```
user@host> show route te-ipv4-prefix-ip 10.10.10.10 extensive
lsdist.0: 298 destinations, 298 routes (298 active, 0 holddown, 0 hidden)
  *IS-IS Preference: 15
    Level: 1
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 298
    Next hop:
    State:<Active NotInstall>
    Local AS: 64496
    Age: 7:58
```



```

Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1000, Flags: 0x40, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0101.0101.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 298
    Next hop:
  State: <Active NotInstall>
  Local AS: 64496
    Age: 7:58
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1000, Flags: 0xe0, Algo: 0>

PREFIX { Node { AS:64496 ISO:0100.0202.0202.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 298
    Next hop:
  State: <Active NotInstall>
  Local AS: 64496
    Age: 7:58
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1000, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0303.0303.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 298
    Next hop:
  State: <Active NotInstall>
  Local AS: 64496
    Age: 7:58
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1000, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0404.0404.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 298
    Next hop:

```

```
State: <Active NotInstall>
Local AS: 64496
Age: 7:58
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1000, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0505.0505.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
*IS-IS Preference: 18
Level: 2
Next hop type: Fictitious, Next hop index: 0
Address: 0xa1a2ac4
Next-hop reference count: 298
Next hop:
State: <Active NotInstall>
Local AS: 64496
Age: 7:58
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1000, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0606.0606.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
*IS-IS Preference: 18
Level: 2
Next hop type: Fictitious, Next hop index: 0
Address: 0xa1a2ac4
Next-hop reference count: 298
Next hop:
State: <Active NotInstall>
Local AS: 64496
Age: 7:58
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1000, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0707.0707.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
*IS-IS Preference: 18
Level: 2
Next hop type: Fictitious, Next hop index: 0
Address: 0xa1a2ac4
Next-hop reference count: 298
Next hop:
State: <Active NotInstall>
Local AS: 64496
Age: 7:58
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1000, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
*IS-IS Preference: 18
Level: 2
Next hop type: Fictitious, Next hop index: 0
```

```

Address: 0xa1a2ac4
Next-hop reference count: 298
Next hop:
State: <Active NotInstall>
Local AS: 64496
Age: 7:58
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1000, Flags: 0x40, Algo: 0

```

show route te-ipv4-prefix-node-iso

```

user@host> show route te-ipv4-prefix-node-iso 0100.0a0a.0a0a.00
Isdist.0: 283 destinations, 283 routes (283 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L1:0
}/1152
      *[IS-IS/15] 00:05:20
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.1.1.1/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:05:20
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.2.2.2/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:05:20
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.3.3.3/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:05:20
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.4.4.4/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:05:20
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.5.5.5/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:05:20
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.6.6.6/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:05:20
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.7.7.7/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:05:20
      Fictitious
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152
      *[IS-IS/18] 00:05:20
      Fictitious

```

show route te-ipv4-prefix-node-iso extensive

```

user@host> show route te-ipv4-prefix-node-iso 0100.0a0a.0a0a.00 extensive
Isdist.0: 283 destinations, 283 routes (283 active, 0 holddown, 0 hidden)
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L1:0

```

```
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 15
    Level: 1
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1000, Flags: 0x40, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.1.1.1/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1001, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.2.2.2/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1002, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.3.3.3/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47
    Validation State: unverified
    Task: IS-IS
    AS path: I
```

```

Prefix SID: 1003, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.4.4.4/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1004, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.5.5.5/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1005, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.6.6.6/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1006, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.7.7.7/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47

```

```

Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1007, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:47
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1000, Flags: 0x40, Algo: 0

```

show route te-ipv4-prefix-node-iso detail

```

user@host> show route te-ipv4-prefix-node-iso 0100.0a0a.0a0a.00 detail
lsdist.0: 283 destinations, 283 routes (283 active, 0 holddown, 0 hidden)
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L1:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 15
    Level: 1
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:54
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1000, Flags: 0x40, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.1.1.1/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:54
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1001, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.2.2.2/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18

```

```

Level: 2
Next hop type: Fictitious, Next hop index: 0
Address: 0xa1a2ac4
Next-hop reference count: 283
Next hop:
State: <Active NotInstall>
Local AS: 64496
Age: 6:54
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1002, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.3.3.3/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
*IS-IS Preference: 18
Level: 2
Next hop type: Fictitious, Next hop index: 0
Address: 0xa1a2ac4
Next-hop reference count: 283
Next hop:
State: <Active NotInstall>
Local AS: 64496
Age: 6:54
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1003, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.4.4.4/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
*IS-IS Preference: 18
Level: 2
Next hop type: Fictitious, Next hop index: 0
Address: 0xa1a2ac4
Next-hop reference count: 283
Next hop:
State: <Active NotInstall>
Local AS: 64496
Age: 6:54
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1004, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.5.5.5/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
*IS-IS Preference: 18
Level: 2
Next hop type: Fictitious, Next hop index: 0
Address: 0xa1a2ac4
Next-hop reference count: 283
Next hop:
State: <Active NotInstall>
Local AS: 64496
Age: 6:54
Validation State: unverified
Task: IS-IS
AS path: I
Prefix SID: 1005, Flags: 0xe0, Algo: 0

```

```
PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.6.6.6/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:54
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1006, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.7.7.7/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:54
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1007, Flags: 0xe0, Algo: 0

PREFIX { Node { AS:64496 ISO:0100.0a0a.0a0a.00 } { IPv4:10.10.10.10/32 } ISIS-L2:0
}/1152 (1 entry, 0 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Fictitious, Next hop index: 0
    Address: 0xa1a2ac4
    Next-hop reference count: 283
    Next hop:
    State: <Active NotInstall>
    Local AS: 64496
    Age: 6:54
    Validation State: unverified
    Task: IS-IS
    AS path: I
    Prefix SID: 1000, Flags: 0x40, Algo: 0
```


show route detail

| | |
|------------------------------------|---|
| List of Syntax | Syntax on page 89 Syntax (EX Series Switches) on page 89 |
| Syntax | show route detail <destination-prefix> <logical-system (all logical-system-name)> |
| Syntax (EX Series Switches) | show route detail <destination-prefix> |
| Release Information | Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 13.2X51-D15 for the QFX Series. Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series. |
| Description | Display detailed information about the active entries in the routing tables. |
| Options | none —Display all active entries in the routing table on all systems. destination-prefix —(Optional) Display active entries for the specified address or range of addresses. logical-system (all logical-system-name) —(Optional) Perform this operation on all logical systems or on a particular logical system. |
| Required Privilege Level | view |
| List of Sample Output | show route detail on page 100 show route detail (with BGP Multipath) on page 106 show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 107 show route label detail (Multipoint LDP with Multicast-Only Fast Reroute) on page 107 |
| Output Fields | Table 5 on page 89 describes the output fields for the show route detail command. Output fields are listed in the approximate order in which they appear. |

Table 5: show route detail Output Fields

| Field Name | Field Description |
|----------------------------|---|
| <i>routing-table-name</i> | Name of the routing table (for example, inet.0). |
| <i>number destinations</i> | Number of destinations for which there are routes in the routing table. |

Table 5: show route detail Output Fields (continued)

| Field Name | Field Description |
|--|--|
| <i>number routes</i> | <p>Number of routes in the routing table and total number of routes in the following states:</p> <ul style="list-style-type: none"> • active (routes that are active) • holddown (routes that are in the pending state before being declared inactive) • hidden (routes that are not used because of a routing policy) |
| <i>route-destination</i> (entry, announced) | <p>Route destination (for example:10.0.0.1/24). The entry value is the number of routes for this destination, and the announced value is the number of routes being announced for this destination. Sometimes the route destination is presented in another format, such as:</p> <ul style="list-style-type: none"> • MPLS-label (for example, 80001). • interface-name (for example, ge-1/0/2). • neighbor-address:control-word-status:encapsulation type:vc-id:source (Layer 2 circuit only; for example, 10.1.1.195:NoCtrlWord:1:1:Local/96). <ul style="list-style-type: none"> • neighbor-address—Address of the neighbor. • control-word-status—Whether the use of the control word has been negotiated for this virtual circuit: NoCtrlWord or CtrlWord. • encapsulation type—Type of encapsulation, represented by a number: (1) Frame Relay DLCI, (2) ATM AAL5 VCC transport, (3) ATM transparent cell transport, (4) Ethernet, (5) VLAN Ethernet, (6) HDLC, (7) PPP, (8) ATM VCC cell transport, (10) ATM VPC cell transport. • vc-id—Virtual circuit identifier. • source—Source of the advertisement: Local or Remote. • source—Source of the advertisement: Local or Remote. |
| <i>label stacking</i> | <p>(Next-to-the-last-hop routing device for MPLS only) Depth of the MPLS label stack, where the label-popping operation is needed to remove one or more labels from the top of the stack. A pair of routes is displayed, because the pop operation is performed only when the stack depth is two or more labels.</p> <ul style="list-style-type: none"> • S=0 route indicates that a packet with an incoming label stack depth of 2 or more exits this routing device with one fewer label (the label-popping operation is performed). • If there is no S= information, the route is a normal MPLS route, which has a stack depth of 1 (the label-popping operation is not performed). |

Table 5: show route detail Output Fields (continued)

| Field Name | Field Description |
|------------------------|--|
| [protocol, preference] | <p>Protocol from which the route was learned and the preference value for the route.</p> <ul style="list-style-type: none"> • +—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • - —A hyphen indicates the last active route. • *—An asterisk indicates that the route is both the active and the last active route. An asterisk before a to line indicates the best subpath to the route. <p>In every routing metric except for the BGP LocalPref attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the LocalPref value in the Preference2 field. For example, if the LocalPref value for Route 1 is 100, the Preference2 value is -101. If the LocalPref value for Route 2 is 155, the Preference2 value is -156. Route 2 is preferred because it has a higher LocalPref value.</p> <p>Preference2 values are signed integers, that is, Preference2 values can be either positive or negative values. However, Junos OS evaluates Preference2 values as unsigned integers that are represented by positive values. Based on the Preference2 values, Junos OS evaluates a preferred route differently in the following scenarios:</p> <ul style="list-style-type: none"> • Both Signed Preference2 values <ul style="list-style-type: none"> • Route A = -101 • Route B = -156 <p>Where both the Preference2 values are signed, Junos OS evaluates only the unsigned value of Preference2 and Route A, which has a lower Preference2 value is preferred.</p> • Unsigned Preference2 values <p>Now consider both unsigned Preference2 values:</p> <ul style="list-style-type: none"> • Route A = 4294967096 • Route B = 200 <p>Here, Junos OS considers the lesser Preference2 value and Route B with a Preference2 value of 200 is preferred because it is less than 4294967096.</p> • Combination of signed and unsigned Preference2 values <p>When Preference2 values of two routes are compared, and for one route the Preference2 is a signed value, and for the other route it is an unsigned value, Junos OS prefers the route with the positive Preference2 value over the negative Preference2 value. For example, consider the following signed and unsigned Preference2 values:</p> <ul style="list-style-type: none"> • Route A = -200 • Route B = 200 <p>In this case, Route B with a Preference2 value of 200 is preferred although this value is greater than -200, because Junos OS evaluates only the unsigned value of the Preference2 value.</p> |
| Level | (IS-IS only). In IS-IS, a single AS can be divided into smaller groups called areas. Routing between areas is organized hierarchically, allowing a domain to be administratively divided into smaller areas. This organization is accomplished by configuring Level 1 and Level 2 intermediate systems. Level 1 systems route within an area. When the destination is outside an area, they route toward a Level 2 system. Level 2 intermediate systems route between areas and toward other ASs. |
| Route Distinguisher | IP subnet augmented with a 64-bit prefix. |
| PMSI | Provider multicast service interface (MVPN routing table). |

Table 5: show route detail Output Fields (continued)

| Field Name | Field Description |
|--|---|
| Next-hop type | Type of next hop. For a description of possible values for this field, see Table 6 on page 95 . |
| Next-hop reference count | Number of references made to the next hop. |
| Flood nexthop branches exceed maximum message | Indicates that the number of flood next-hop branches exceeded the system limit of 32 branches, and only a subset of the flood next-hop branches were installed in the kernel. |
| Source | IP address of the route source. |
| Next hop | Network layer address of the directly reachable neighboring system. |
| via | <p>Interface used to reach the next hop. If there is more than one interface available to the next hop, the name of the interface that is actually used is followed by the word Selected. This field can also contain the following information:</p> <ul style="list-style-type: none"> • Weight—Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible. • Balance—Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a routing device is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing. |
| Label-switched-path lsp-path-name | Name of the LSP used to reach the next hop. |
| Label operation | MPLS label and operation occurring at this routing device. The operation can be pop (where a label is removed from the top of the stack), push (where another label is added to the label stack), or swap (where a label is replaced by another label). |
| Interface | (Local only) Local interface name. |
| Protocol next hop | Network layer address of the remote routing device that advertised the prefix. This address is used to derive a forwarding next hop. |
| Indirect next hop | Index designation used to specify the mapping between protocol next hops, tags, kernel export policy, and the forwarding next hops. |
| State | State of the route (a route can be in more than one state). See Table 7 on page 97 . |
| Local AS | AS number of the local routing device. |
| Age | How long the route has been known. |
| AIGP | Accumulated interior gateway protocol (AIGP) BGP attribute. |

Table 5: show route detail Output Fields (continued)

| Field Name | Field Description |
|--------------------------|---|
| Metric | Cost value of the indicated route. For routes within an AS, the cost is determined by IGP and the individual protocol metrics. For external routes, destinations, or routing domains, the cost is determined by a preference value. |
| MED-plus-IGP | Metric value for BGP path selection to which the IGP cost to the next-hop destination has been added. |
| TTL-Action | For MPLS LSPs, state of the TTL propagation attribute. Can be enabled or disabled for all RSVP-signaled and LDP-signaled LSPs or for specific VRF routing instances. For sample output, see show route table . |
| Task | Name of the protocol that has added the route. |
| Announcement bits | The number of BGP peers or protocols to which Junos OS has announced this route, followed by the list of the recipients of the announcement. Junos OS can also announce the route to the KRT for installing the route into the Packet Forwarding Engine, to a resolve tree, a L2 VC, or even a VPN. For example, n-Resolve inet indicates that the specified route is used for route resolution for next hops found in the routing table. <ul style="list-style-type: none"> n—An index used by Juniper Networks customer support only. |
| AS path | <p>AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated:</p> <ul style="list-style-type: none"> I—IGP. E—EGP. Recorded—The AS path is recorded by the sample process (sampled). ?—Incomplete; typically, the AS path was aggregated. <p>When AS path numbers are included in the route, the format is as follows:</p> <ul style="list-style-type: none"> []—Brackets enclose the number that precedes the AS path. This number represents the number of ASs present in the AS path, when calculated as defined in RFC 4271. This value is used in the AS-path merge process, as defined in RFC 4893. []—If more than one AS number is configured on the routing device, or if AS path prepending is configured, brackets enclose the local AS number associated with the AS path. { }—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. ()—Parentheses enclose a confederation. ([])—Parentheses and brackets enclose a confederation set. <p>NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.</p> |

Table 5: show route detail Output Fields (continued)

| Field Name | Field Description |
|-------------------------|---|
| validation-state | <p>(BGP-learned routes) Validation status of the route:</p> <ul style="list-style-type: none"> • Invalid—Indicates that the prefix is found, but either the corresponding AS received from the EBGP peer is not the AS that appears in the database, or the prefix length in the BGP update message is longer than the maximum length permitted in the database. • Unknown—Indicates that the prefix is not among the prefixes or prefix ranges in the database. • Unverified—Indicates that the origin of the prefix is not verified against the database. This is because the database got populated and the validation is not called for in the BGP import policy, although origin validation is enabled, or the origin validation is not enabled for the BGP peers. • Valid—Indicates that the prefix and autonomous system pair are found in the database. |
| ORR Generation-ID | Displays the optimal route reflection (ORR) generation identifier. ISIS and OSPF interior gateway protocol (IGP) updates filed whenever any of the corresponding ORR route has its metric valued changed, or if the ORR route is added or deleted. |
| FECs bound to route | Point-to-multipoint root address, multicast source address, and multicast group address when multipoint LDP (M-LDP) inband signaling is configured. |
| Primary Upstream | When multipoint LDP with multicast-only fast reroute (MoFRR) is configured, the primary upstream path. MoFRR transmits a multicast join message from a receiver toward a source on a primary path, while also transmitting a secondary multicast join message from the receiver toward the source on a backup path. |
| RPF Nexthops | When multipoint LDP with MoFRR is configured, the reverse-path forwarding (RPF) next-hop information. Data packets are received from both the primary path and the secondary paths. The redundant packets are discarded at topology merge points due to the RPF checks. |
| Label | Multiple MPLS labels are used to control MoFRR stream selection. Each label represents a separate route, but each references the same interface list check. Only the primary label is forwarded while all others are dropped. Multiple interfaces can receive packets using the same label. |
| weight | Value used to distinguish MoFRR primary and backup routes. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible. |
| VC Label | MPLS label assigned to the Layer 2 circuit virtual connection. |
| MTU | Maximum transmission unit (MTU) of the Layer 2 circuit. |
| VLAN ID | VLAN identifier of the Layer 2 circuit. |
| Prefixes bound to route | Forwarding equivalent class (FEC) bound to this route. Applicable only to routes installed by LDP. |
| Communities | Community path attribute for the route. See Table 8 on page 99 for all possible values for this field. |
| Layer2-info: encaps | Layer 2 encapsulation (for example, VPLS). |
| control flags | Control flags: none or Site Down . |
| mtu | Maximum transmission unit (MTU) information. |

Table 5: show route detail Output Fields (continued)

| Field Name | Field Description |
|--|--|
| Label-Base, range | First label in a block of labels and label block size. A remote PE routing device uses this first label when sending traffic toward the advertising PE routing device. |
| status vector | Layer 2 VPN and VPLS network layer reachability information (NLRI). |
| Accepted Multipath | Current active path when BGP multipath is configured. |
| Accepted LongLivedStale | The LongLivedStale flag indicates that the route was marked LLGR-stale by this router, as part of the operation of LLGR receiver mode. Either this flag or the LongLivedStaleImport flag may be displayed for a route. Neither of these flags are displayed at the same time as the Stale (ordinary GR stale) flag. |
| Accepted LongLivedStaleImport | <p>The LongLivedStaleImport flag indicates that the route was marked LLGR-stale when it was received from a peer, or by import policy. Either this flag or the LongLivedStale flag may be displayed for a route. Neither of these flags are displayed at the same time as the Stale (ordinary GR stale) flag.</p> <p>Accept all received BGP long-lived graceful restart (LLGR) and LLGR stale routes learned from configured neighbors and import into the inet.0 routing table</p> |
| ImportAccepted LongLivedStaleImport | <p>Accept all received BGP long-lived graceful restart (LLGR) and LLGR stale routes learned from configured neighbors and imported into the inet.0 routing table</p> <p>The LongLivedStaleImport flag indicates that the route was marked LLGR-stale when it was received from a peer, or by import policy.</p> |
| Accepted MultipathContrib | Path currently contributing to BGP multipath. |
| Localpref | Local preference value included in the route. |
| Router ID | BGP router ID as advertised by the neighbor in the open message. |
| Primary Routing Table | In a routing table group, the name of the primary routing table in which the route resides. |
| Secondary Tables | In a routing table group, the name of one or more secondary tables in which the route resides. |

[Table 6 on page 95](#) describes all possible values for the Next-hop Types output field.

Table 6: Next-hop Types Output Field Values

| Next-Hop Type | Description |
|--------------------------|---------------------|
| Broadcast (bcast) | Broadcast next hop. |
| Deny | Deny next hop. |
| Discard | Discard next hop. |

Table 6: Next-hop Types Output Field Values (continued)

| Next-Hop Type | Description |
|---------------------------------|--|
| Flood | Flood next hop. Consists of components called branches, up to a maximum of 32 branches. Each flood next-hop branch sends a copy of the traffic to the forwarding interface. Used by point-to-multipoint RSVP, point-to-multipoint LDP, point-to-multipoint CCC, and multicast. |
| Hold | Next hop is waiting to be resolved into a unicast or multicast type. |
| Indexed (idxd) | Indexed next hop. |
| Indirect (indr) | Used with applications that have a protocol next hop address that is remote. You are likely to see this next-hop type for internal BGP (IBGP) routes when the BGP next hop is a BGP neighbor that is not directly connected. |
| Interface | Used for a network address assigned to an interface. Unlike the router next hop, the interface next hop does not reference any specific node on the network. |
| Local (locl) | Local address on an interface. This next-hop type causes packets with this destination address to be received locally. |
| Multicast (mcst) | Wire multicast next hop (limited to the LAN). |
| Multicast discard (mdsc) | Multicast discard. |
| Multicast group (mgrp) | Multicast group member. |
| Receive (recv) | Receive. |
| Reject (rjct) | Discard. An ICMP unreachable message was sent. |
| Resolve (rslv) | Resolving next hop. |
| Routed multicast (mcrt) | Regular multicast next hop. |
| Router | <p>A specific node or set of nodes to which the routing device forwards packets that match the route prefix.</p> <p>To qualify as next-hop type router, the route must meet the following criteria:</p> <ul style="list-style-type: none"> • Must not be a direct or local subnet for the routing device. • Must have a next hop that is directly connected to the routing device. |
| Table | Routing table next hop. |

Table 6: Next-hop Types Output Field Values (continued)

| Next-Hop Type | Description |
|----------------|---|
| Unicast (ucst) | Unicast. |
| Unilist (ulst) | List of unicast next hops. A packet sent to this next hop goes to any next hop in the list. |

Table 7 on page 97 describes all possible values for the State output field. A route can be in more than one state (for example, **<Active NoReadvrt Int Ext>**).

Table 7: State Output Field Values

| Value | Description |
|---------------------------------------|--|
| Accounting | Route needs accounting. |
| Active | Route is active. |
| Always Compare MED | Path with a lower multiple exit discriminator (MED) is available. |
| AS path | Shorter AS path is available. |
| Cisco Non-deterministic MED selection | Cisco nondeterministic MED is enabled, and a path with a lower MED is available. |
| Clone | Route is a clone. |
| Cluster list length | Length of cluster list sent by the route reflector. |
| Delete | Route has been deleted. |
| Ex | Exterior route. |
| Ext | BGP route received from an external BGP neighbor. |
| FlashAll | Forces all protocols to be notified of a change to any route, active or inactive, for a prefix. When not set, protocols are informed of a prefix only when the active route changes. |
| Hidden | Route not used because of routing policy. |
| IfCheck | Route needs forwarding RPF check. |
| IGP metric | Path through next hop with lower IGP metric is available. |
| Inactive reason | Flags for this route, which was not selected as best for a particular destination. |
| Initial | Route being added. |

Table 7: State Output Field Values (continued)

| Value | Description |
|--|---|
| Int | Interior route. |
| Int Ext | BGP route received from an internal BGP peer or a BGP confederation peer. |
| Interior > Exterior > Exterior via Interior | Direct, static, IGP, or EBGp path is available. |
| Local Preference | Path with a higher local preference value is available. |
| Martian | Route is a martian (ignored because it is obviously invalid). |
| MartianOK | Route exempt from martian filtering. |
| Next hop address | Path with lower metric next hop is available. |
| No difference | Path from neighbor with lower IP address is available. |
| NoReadvrt | Route not to be advertised. |
| NotBest | Route not chosen because it does not have the lowest MED. |
| Not Best in its group | Incoming BGP AS is not the best of a group (only one AS can be the best). |
| NotInstall | Route not to be installed in the forwarding table. |
| Number of gateways | Path with a greater number of next hops is available. |
| Origin | Path with a lower origin code is available. |
| Pending | Route pending because of a hold-down configured on another route. |
| Programmed | Route installed programmatically by on-box or off-box applications using API. |
| Release | Route scheduled for release. |
| RIB preference | Route from a higher-numbered routing table is available. |
| Route Distinguisher | 64-bit prefix added to IP subnets to make them unique. |
| Route Metric or MED comparison | Route with a lower metric or MED is available. |
| Route Preference | Route with lower preference value is available |
| Router ID | Path through a neighbor with lower ID is available. |

Table 7: State Output Field Values (continued)

| Value | Description |
|----------------|---|
| Secondary | Route not a primary route. |
| Unusable path | Path is not usable because of one of the following conditions: <ul style="list-style-type: none"> The route is damped. The route is rejected by an import policy. The route is unresolved. |
| Update source | Last tiebreaker is the lowest IP address value. |
| ProtectionCand | Indicates paths requesting protection. |
| ProtectionPath | Indicates the route entry that can be used as a protection path. |

Table 8 on page 99 describes the possible values for the Communities output field.

Table 8: Communities Output Field Values

| Value | Description |
|---|---|
| <i>area-number</i> | 4 bytes, encoding a 32-bit area number. For AS-external routes, the value is 0 . A nonzero value identifies the route as internal to the OSPF domain, and as within the identified area. Area numbers are relative to a particular OSPF domain. |
| bandwidth: local AS number:link-bandwidth-number | Link-bandwidth community value used for unequal-cost load balancing. When BGP has several candidate paths available for multipath purposes, it does not perform unequal-cost load balancing according to the link-bandwidth community unless all candidate paths have this attribute. |
| domain-id | Unique configurable number that identifies the OSPF domain. |
| domain-id-vendor | Unique configurable number that further identifies the OSPF domain. |
| <i>link-bandwidth-number</i> | Link-bandwidth number: from 0 through 4,294,967,295 (bytes per second). |
| <i>local AS number</i> | Local AS number: from 1 through 65,535 . |
| <i>options</i> | 1 byte. Currently this is only used if the route type is 5 or 7 . Setting the least significant bit in the field indicates that the route carries a type 2 metric. |
| origin | (Used with VPNs) Identifies where the route came from. |
| <i>ospf-route-type</i> | 1 byte, encoded as 1 or 2 for intra-area routes (depending on whether the route came from a type 1 or a type 2 LSA); 3 for summary routes; 5 for external routes (area number must be 0); 7 for NSSA routes; or 129 for sham link endpoint addresses. |
| route-type-vendor | Displays the area number, OSPF route type, and option of the route. This is configured using the BGP extended community attribute 0x8000 . The format is area-number:ospf-route-type:options . |

Table 8: Communities Output Field Values (continued)

| Value | Description |
|--------------------------------------|--|
| rte-type | Displays the area number, OSPF route type, and option of the route. This is configured using the BGP extended community attribute 0x0306 . The format is area-number:ospf-route-type:options . |
| target | Defines which VPN the route participates in; target has the format 32-bit IP address:16-bit number . For example, 10.19.0.0:100. |
| unknown IANA | Incoming IANA codes with a value between 0x1 and 0x7fff . This code of the BGP extended community attribute is accepted, but it is not recognized. |
| unknown OSPF vendor community | Incoming IANA codes with a value above 0x8000 . This code of the BGP extended community attribute is accepted, but it is not recognized. |

Sample Output

show route detail

```

user@host> show route detail

inet.0: 22 destinations, 23 routes (21 active, 0 holddown, 1 hidden)
10.10.0.0/16 (1 entry, 1 announced)
  *Static Preference: 5
    Next-hop reference count: 29
    Next hop: 192.168.71.254 via fxp0.0, selected
    State: <Active NoReadvrt Int Ext>
    Local AS: 69
    Age: 1:31:43
    Task: RT
    Announcement bits (2): 0-KRT 3-Resolve tree 2
    AS path: I

10.31.1.0/30 (2 entries, 1 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 2
    Next hop: via so-0/3/0.0, selected
    State: <Active Int>
    Local AS: 69
    Age: 1:30:17
    Task: IF
    Announcement bits (1): 3-Resolve tree 2
    AS path: I
  OSPF Preference: 10
    Next-hop reference count: 1
    Next hop: via so-0/3/0.0, selected
    State: <Int>
    Inactive reason: Route Preference
    Local AS: 69
    Age: 1:30:17 Metric: 1
    ORR Generation-ID: 1
  Area: 0.0.0.0
    Task: OSPF
    AS path: I

```

```

10.31.1.1/32 (1 entry, 1 announced)
  *Local Preference: 0
    Next hop type: Local
    Next-hop reference count: 7
    Interface: so-0/3/0.0
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:30:20
    Task: IF
    Announcement bits (1): 3-Resolve tree 2
    AS path: I

...

10.31.2.0/30 (1 entry, 1 announced)
  *OSPF Preference: 10
    Next-hop reference count: 9
    Next hop: via so-0/3/0.0
    Next hop: 10.31.1.6 via ge-3/1/0.0, selected
    State: <Active Int>
    Local AS: 69
    Age: 1:29:56 Metric: 2
    Area: 0.0.0.0
    ORR Generation-ID: 1
  Task: OSPF
    Announcement bits (2): 0-KRT 3-Resolve tree 2
    AS path: I

...

172.16.233.2/32 (1 entry, 1 announced)
  *PIM Preference: 0
    Next-hop reference count: 18
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:31:45
    Task: PIM Recv
    Announcement bits (2): 0-KRT 3-Resolve tree 2
    AS path: I

...

172.16.233.22/32 (1 entry, 1 announced)
  *IGMP Preference: 0
    Next-hop reference count: 18
    State: <Active NoReadvrt Int>
    Local AS: 69
    Age: 1:31:43
    Task: IGMP
    Announcement bits (2): 0-KRT 3-Resolve tree 2
    AS path: I

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

10.255.70.103/32 (1 entry, 1 announced)
  State: <FlashAll>
  *RSVP Preference: 7
    Next-hop reference count: 6
    Next hop: 10.31.1.6 via ge-3/1/0.0 weight 0x1, selected
    Label-switched-path green-r1-r3
    Label operation: Push 100096

```

```

State: <Active Int>
Local AS: 69
Age: 1:25:49 Metric: 2
Task: RSVP
Announcement bits (2): 1-Resolve tree 1 2-Resolve tree 2
AS path: I

10.255.71.238/32 (1 entry, 1 announced)
State: <FlashAll>
*RSVP Preference: 7
Next-hop reference count: 6
Next hop: via so-0/3/0.0 weight 0x1, selected
Label-switched-path green-r1-r2
State: <Active Int>
Local AS: 69
Age: 1:25:49 Metric: 1
Task: RSVP
Announcement bits (2): 1-Resolve tree 1 2-Resolve tree 2
AS path: I

private__inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

47.0005.80ff.f800.0000.0108.0001.0102.5507.1052/152 (1 entry, 0 announced)
*Direct Preference: 0
Next hop type: Interface
Next-hop reference count: 1
Next hop: via lo0.0, selected
State: <Active Int>
Local AS: 69
Age: 1:31:44
Task: IF
AS path: I

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
0 (1 entry, 1 announced)
*MPLS Preference: 0
Next hop type: Receive
Next-hop reference count: 6
State: <Active Int>
Local AS: 69
Age: 1:31:45 Metric: 1
Task: MPLS
Announcement bits (1): 0-KRT
AS path: I

...

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

299840 (1 entry, 1 announced)
TSI:
KRT in-kerne 299840 /52 -> {indirect(1048575)}
*RSVP Preference: 7/2
Next hop type: Flood
Address: 0x9174a30
Next-hop reference count: 4
Next hop type: Router, Next hop index: 798
Address: 0x9174c28
Next-hop reference count: 2

```

```

Next hop: 172.16.0.2 via lt-1/2/0.9 weight 0x1
Label-switched-path R2-to-R4-2p2mp
Label operation: Pop
Next hop type: Router, Next hop index: 1048574
Address: 0x92544f0
Next-hop reference count: 2
Next hop: 172.16.0.2 via lt-1/2/0.7 weight 0x1
Label-switched-path R2-to-R200-p2mp
Label operation: Pop
Next hop: 172.16.0.2 via lt-1/2/0.5 weight 0x8001
Label operation: Pop
State: <Active Int>
Age: 1:29      Metric: 1
Task: RSVP
Announcement bits (1): 0-KRT
AS path: I...

800010 (1 entry, 1 announced)
  *VPLS Preference: 7
    Next-hop reference count: 2
    Next hop: via vt-3/2/0.32769, selected
    Label operation: Pop
    State: <Active Int>
    Age: 1:29:30
    Task: Common L2 VC
    Announcement bits (1): 0-KRT
    AS path: I

vt-3/2/0.32769 (1 entry, 1 announced)
  *VPLS Preference: 7
    Next-hop reference count: 2
    Next hop: 10.31.1.6 via ge-3/1/0.0 weight 0x1, selected
    Label-switched-path green-r1-r3
    Label operation: Push 800012, Push 100096(top)
    Protocol next hop: 10.255.70.103
    Push 800012
    Indirect next hop: 87272e4 1048574
    State: <Active Int>
    Age: 1:29:30      Metric2: 2
    Task: Common L2 VC
    Announcement bits (2): 0-KRT 1-Common L2 VC
    AS path: I
    Communities: target:11111:1 Layer2-info: encaps:VPLS,
    control flags:, mtu: 0

inet6.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)

abcd::10:255:71:52/128 (1 entry, 0 announced)
  *Direct Preference: 0
    Next hop type: Interface
    Next-hop reference count: 1
    Next hop: via lo0.0, selected
    State: <Active Int>
    Local AS: 69
    Age: 1:31:44
    Task: IF
    AS path: I

fe80::280:42ff:fe10:f179/128 (1 entry, 0 announced)
  *Direct Preference: 0
    Next hop type: Interface

```

```

        Next-hop reference count: 1
        Next hop: via lo0.0, selected
        State: <Active NoReadvrt Int>
        Local AS:    69
        Age: 1:31:44
        Task: IF
        AS path: I

ff02::2/128 (1 entry, 1 announced)
  *PIM    Preference: 0
          Next-hop reference count: 18
          State: <Active NoReadvrt Int>
          Local AS:    69
          Age: 1:31:45
          Task: PIM Recv6
          Announcement bits (1): 0-KRT
          AS path: I

ff02::d/128 (1 entry, 1 announced)
  *PIM    Preference: 0
          Next-hop reference count: 18
          State: <Active NoReadvrt Int>
          Local AS:    69
          Age: 1:31:45
          Task: PIM Recv6
          Announcement bits (1): 0-KRT
          AS path: I

ff02::16/128 (1 entry, 1 announced)
  *MLD    Preference: 0
          Next-hop reference count: 18
          State: <Active NoReadvrt Int>
          Local AS:    69
          Age: 1:31:43
          Task: MLD
          Announcement bits (1): 0-KRT
          AS path: I

private.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

fe80::280:42ff:fe10:f179/128 (1 entry, 0 announced)
  *Direct Preference: 0
          Next hop type: Interface
          Next-hop reference count: 1
          Next hop: via lo0.16385, selected
          State: <Active NoReadvrt Int>
          Age: 1:31:44
          Task: IF
          AS path: I

green.l2vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)

10.255.70.103:1:3:1/96 (1 entry, 1 announced)
  *BGP    Preference: 170/-101
          Route Distinguisher: 10.255.70.103:1
          Next-hop reference count: 7
          Source: 10.255.70.103
          Protocol next hop: 10.255.70.103
          Indirect next hop: 2 no-forward
          State: <Secondary Active Int Ext>
          Local AS:    69 Peer AS:    69
```



```

Age: 1:25:49    Metric2: 1
AIGP 210
Task: BGP_69.10.255.70.103+179
Announcement bits (1): 0-green-l2vpn
AS path: I
Communities: target:11111:1 Layer2-info: encaps:VPLS,
control flags:, mtu: 0
Label-base: 800008, range: 8
Localpref: 100
Router ID: 10.255.70.103
Primary Routing Table bgp.l2vpn.0

10.255.71.52:1:1:1/96 (1 entry, 1 announced)
  *L2VPN Preference: 170/-1
    Next-hop reference count: 5
    Protocol next hop: 10.255.71.52
    Indirect next hop: 0 -
    State: <Active Int Ext>
    Age: 1:31:40    Metric2: 1
    Task: green-l2vpn
    Announcement bits (1): 1-BGP.0.0.0.0+179
    AS path: I
    Communities: Layer2-info: encaps:VPLS, control flags:Site-Down,
    mtu: 0
    Label-base: 800016, range: 8, status-vector: 0x9F

10.255.71.52:1:5:1/96 (1 entry, 1 announced)
  *L2VPN Preference: 170/-101
    Next-hop reference count: 5
    Protocol next hop: 10.255.71.52
    Indirect next hop: 0 -
    State: <Active Int Ext>
    Age: 1:31:40    Metric2: 1
    Task: green-l2vpn
    Announcement bits (1): 1-BGP.0.0.0.0+179
    AS path: I
    Communities: Layer2-info: encaps:VPLS, control flags:, mtu: 0
    Label-base: 800008, range: 8, status-vector: 0x9F

...

l2circuit.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.245.255.63:CtrlWord:4:3:Local/96 (1 entry, 1 announced)
  *L2CKT Preference: 7
    Next hop: via so-1/1/2.0 weight 1, selected
    Label-switched-path my-lsp
    Label operation: Push 100000[0]
    Protocol next hop: 10.245.255.63 Indirect next hop: 86af000 296
    State: <Active Int>
    Local AS: 99
    Age: 10:21
    Task: l2 circuit
    Announcement bits (1): 0-LDP
    AS path: I
    VC Label 100000, MTU 1500, VLAN ID 512

inet.0: 45 destinations, 47 routes (44 active, 0 holddown, 1 hidden)
1.1.1.3/32 (1 entry, 1 announced)
  *IS-IS Preference: 18
    Level: 2
    Next hop type: Router, Next hop index: 580

```

```

Address: 0x9db6ed0
Next-hop reference count: 8
Next hop: 10.1.1.6 via lt-1/0/10.5, selected
Session Id: 0x18a
State: <Active Int>
Local AS: 2
Age: 1:32 Metric: 10
Validation State: unverified
ORR Generation-ID: 1
Task: IS-IS
Announcement bits (3): 0-KRT 5-Resolve tree 4 6-Resolve_IGP_FRR
task
AS path: I

inet.0: 61 destinations, 77 routes (61 active, 1 holddown, 0 hidden)
1.1.1.1/32 (2 entries, 1 announced)
*OSPF Preference: 10
Next hop type: Router, Next hop index: 673
Address: 0xc008830
Next-hop reference count: 3
Next hop: 10.1.1.1 via ge-0/0/2.0, selected
Session Id: 0x1b7
State: <Active Int>
Local AS: 1
Age: 3:06:59 Metric: 100
Validation State: unverified
ORR Generation-ID: 1
Area: 0.0.0.0
Task: OSPF
Announcement bits (2): 1-KRT 9-Resolve tree 2
AS path: I

```

show route detail (with BGP Multipath)

```

user@host> show route detail

10.1.1.8/30 (2 entries, 1 announced)
*BGP Preference: 170/-101
Next hop type: Router, Next hop index: 262142
Address: 0x901a010
Next-hop reference count: 2
Source: 10.1.1.2
Next hop: 10.1.1.2 via ge-0/3/0.1, selected
Next hop: 10.1.1.6 via ge-0/3/0.5
State: <Active Ext>
Local AS: 1 Peer AS: 2
Age: 5:04:43
Validation State: unverified
Task: BGP_2.10.1.1.2+59955
Announcement bits (1): 0-KRT
AS path: 2 I
Accepted Multipath
Localpref: 100
Router ID: 172.16.1.2
BGP Preference: 170/-101
Next hop type: Router, Next hop index: 678
Address: 0x8f97520
Next-hop reference count: 9
Source: 10.1.1.6
Next hop: 10.1.1.6 via ge-0/3/0.5, selected
State: <NotBest Ext>

```

```

Inactive reason: Not Best in its group - Active preferred
Local AS:      1 Peer AS:      2
Age: 5:04:43
Validation State: unverified
Task: BGP_2.10.1.1.6+58198
AS path: 2 I
Accepted MultipathContrib
Localpref: 100
Router ID: 172.16.1.3

```

show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```

user@host> show route label 299872 detail
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
299872 (1 entry, 1 announced)
  *LDP    Preference: 9
          Next hop type: Flood
          Next-hop reference count: 3
          Address: 0x9097d90
          Next hop: via vt-0/1/0.1
          Next-hop index: 661
          Label operation: Pop
          Address: 0x9172130
          Next hop: via so-0/0/3.0
          Next-hop index: 654
          Label operation: Swap 299872
          State: **Active Int>
          Local AS: 1001
          Age: 8:20      Metric: 1
          Task: LDP
          Announcement bits (1): 0-KRT
          AS path: I
          FECs bound to route: P2MP root-addr 10.255.72.166, grp 232.1.1.1,
src 192.168.142.2

```

show route label detail (Multipoint LDP with Multicast-Only Fast Reroute)

```

user@host> show route label 301568 detail
mpls.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
301568 (1 entry, 1 announced)
  *LDP    Preference: 9
          Next hop type: Flood
          Address: 0x2735208
          Next-hop reference count: 3
          Next hop type: Router, Next hop index: 1397
          Address: 0x2735d2c
          Next-hop reference count: 3
          Next hop: 1.3.8.2 via ge-1/2/22.0
          Label operation: Pop
          Load balance label: None;
          Next hop type: Router, Next hop index: 1395
          Address: 0x2736290
          Next-hop reference count: 3
          Next hop: 1.3.4.2 via ge-1/2/18.0
          Label operation: Pop
          Load balance label: None;
          State: <Active Int AckRequest MulticastRPF>
          Local AS: 10
          Age: 54:05      Metric: 1

```

```
Validation State: unverified
Task: LDP
Announcement bits (1): 0-KRT
AS path: I
FECs bound to route: P2MP root-addr 172.16.1.1, grp: 232.1.1.1,
src: 192.168.219.11
Primary Upstream : 172.16.1.3:0--172.16.1.2:0
  RPF Nexthops :
    ge-1/2/15.0, 1.2.94.1, Label: 301568, weight: 0x1
    ge-1/2/14.0, 1.2.3.1, Label: 301568, weight: 0x1
Backup Upstream : 172.16.1.3:0--172.16.1.6:0
  RPF Nexthops :
    ge-1/2/20.0, 1.2.96.1, Label: 301584, weight: 0xffffe
    ge-1/2/19.0, 1.3.6.1, Label: 301584, weight: 0xffffe
```