

# EVPNs on EX9200 Ethernet Switches

Release  
15.1



---

Modified: 2015-06-02

Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://www.juniper.net)

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

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

*EVPNs on EX9200 Ethernet Switches*

15.1

Copyright © 2015, Juniper Networks, Inc.  
All rights reserved.

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

#### YEAR 2000 NOTICE

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

#### END USER LICENSE AGREEMENT

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

# Table of Contents

	About the Documentation . . . . .	ix
	Documentation and Release Notes . . . . .	ix
	Supported Platforms . . . . .	ix
	Using the Examples in This Manual . . . . .	ix
	Merging a Full Example . . . . .	x
	Merging a Snippet . . . . .	x
	Documentation Conventions . . . . .	xi
	Documentation Feedback . . . . .	xiii
	Requesting Technical Support . . . . .	xiii
	Self-Help Online Tools and Resources . . . . .	xiii
	Opening a Case with JTAC . . . . .	xiv
<b>Part 1</b>	<b>Understanding Ethernet VPNs</b>	
<b>Chapter 1</b>	<b>Overview . . . . .</b>	<b>3</b>
	EVPN Overview for Switches . . . . .	3
<b>Chapter 2</b>	<b>EVPN Standards . . . . .</b>	<b>5</b>
	Supported EVPN Standards . . . . .	5
<b>Chapter 3</b>	<b>Distributing Routes in VPNs . . . . .</b>	<b>7</b>
	Configuring an IGP on the PE and P Routers on EX9200 Switches . . . . .	7
	Configuring IBGP Sessions Between PE Routers in VPNs on EX9200 Switches . . . . .	7
	Configuring a Signaling Protocol and LSPs for VPNs on EX9200 Switches . . . . .	9
	Using LDP for VPN Signaling . . . . .	9
<b>Part 2</b>	<b>Configuring EVPNs</b>	
<b>Chapter 4</b>	<b>Configuring Routing Instances for EVPN . . . . .</b>	<b>13</b>
	Configuring EVPN Routing Instances on EX9200 Switches . . . . .	13
	Tracing EVPN Traffic and Operations . . . . .	15
<b>Chapter 5</b>	<b>Configuring Integrated Bridging and Routing Using EVPNs . . . . .</b>	<b>17</b>
	An EVPN with IRB Solution on EX9200 Switches Overview . . . . .	17
	Need for an EVPN IRB Solution . . . . .	18
	Implementing the EVPN IRB Solution . . . . .	18
	Benefits of Implementing the EVPN IRB Solution . . . . .	20
	Gateway MAC and IP Synchronization . . . . .	20
	Layer 3 VPN Interworking . . . . .	20
	Configuring an EVPN with IRB Solution on EX9200 Switches . . . . .	22
	Example: Configuring an EVPN with IRB Solution on EX9200 Switches . . . . .	23

<b>Part 3</b>	<b>Configuration Statements and Operational Commands</b>	
<b>Chapter 6</b>	<b>Configuration Statements</b> . . . . .	<b>37</b>
	encapsulation (Physical Interface) . . . . .	38
	evpn . . . . .	43
	extended-vlan-list . . . . .	44
	instance-type . . . . .	45
	interface (EVPN Routing Instances) . . . . .	47
	interface (Routing Instances) . . . . .	48
	interface-mac-limit (VPLS) . . . . .	49
	mac-statistics . . . . .	50
	mac-table-size . . . . .	51
	no-mac-learning . . . . .	52
	packet-action . . . . .	54
	routing-instances . . . . .	56
	traceoptions (Protocols EVPN) . . . . .	57
	vlan-id (routing instance) . . . . .	59
<b>Chapter 7</b>	<b>Operational Commands</b> . . . . .	<b>61</b>
	show evpn database . . . . .	62
	show evpn flood . . . . .	63
	show evpn instance . . . . .	64
	show evpn mac-table . . . . .	69
	show evpn peer-gateway-macs . . . . .	70
	show evpn statistics . . . . .	71

# List of Figures

Part 2	Configuring EVPNs	
Chapter 5	Configuring Integrated Bridging and Routing Using EVPNs . . . . .	17
	Figure 1: Inter-Subnet Traffic Forwarding . . . . .	19



# List of Tables

	<b>About the Documentation . . . . .</b>	<b>ix</b>
	Table 1: Notice Icons . . . . .	xi
	Table 2: Text and Syntax Conventions . . . . .	xi
<b>Part 3</b>	<b>Configuration Statements and Operational Commands</b>	
<b>Chapter 7</b>	<b>Operational Commands . . . . .</b>	<b>61</b>
	Table 3: show evpn instance Output Fields . . . . .	64





# About the Documentation

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

## Documentation and Release Notes

---

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

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

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

## Supported Platforms

---

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

- EX Series

## 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 the *CLI User Guide*.

## Documentation Conventions

Table 1 on page xi 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 xi defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  user@host> <b>configure</b>

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&gt; show chassis alarms</pre> <p>No alarms currently active</p>
<i>Italic text like this</i>	<ul style="list-style-type: none"> <li>Introduces or emphasizes important new terms.</li> <li>Identifies guide names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul>	<ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS CLI User Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	<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"> <li>To configure a stub area, include the <b>stub</b> statement at the <code>[edit protocols ospf area area-id]</code> hierarchy level.</li> <li>The console port is labeled <b>CONSOLE</b>.</li> </ul>
< > (angle brackets)	Encloses optional keywords or variables.	<b>stub</b> <default-metric <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	<b>broadcast   multicast</b> <i>(string1   string2   string3)</i>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	<b>community name members [ community-ids ]</b>
Indentation and braces ( { } )	Identifies a level in the configuration hierarchy.	<pre>[edit] routing-options {   static {     route default {       nexthop address;       retain;     }   } }</pre>
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
<b>GUI Conventions</b>		
<b>Bold text like this</b>	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> <li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li> <li>To cancel the configuration, click <b>Cancel</b>.</li> </ul>

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 <b>Protocols&gt;Ospf</b> .

## Documentation Feedback

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

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

## Requesting Technical Support

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

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

## Self-Help Online Tools and Resources

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

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>

- Download the latest versions of software and review release notes:  
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:  
<http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum:  
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

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

## Opening a Case with JTAC

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

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

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

## PART 1

# Understanding Ethernet VPNs

- [Overview on page 3](#)
- [EVPN Standards on page 5](#)





## CHAPTER 1

# Overview

- [EVPN Overview for Switches on page 3](#)

### EVPN Overview for Switches

---

An Ethernet VPN (EVPN) enables you to connect a group of dispersed customer sites using a Layer 2 virtual bridge. As with other types of VPNs, an EVPN is comprised of customer edge (CE) devices (host, router, or switch) connected to provider edge (PE) devices. The PE devices can include an MPLS edge switch (MES) that acts at the edge of the MPLS infrastructure. For the initial deployment of EVPNs using Juniper Networks equipment, you can configure an EX9200 switch to act as an MES. You can deploy multiple EVPNs within the network, each providing network connectivity to customers while ensuring that traffic sharing that network remains private.

The MESs are interconnected within the network by using label-switched paths (LSPs). The MPLS infrastructure allows you to take advantage of the MPLS functionality provided by the Junos operating system (Junos OS), including fast reroute, node and link protection, and standby secondary paths. For EVPNs, learning between MESs takes place in the control plane rather than in the data plane (as is the case with traditional network bridging). The control plane provides greater control over the learning process, allowing you to restrict which devices discover information about the network. You can also apply policies on the MESs, allowing you to carefully control how network information is distributed and processed. EVPNs utilize the BGP control plane infrastructure, providing greater scale and the ability to isolate groups of devices (hosts, servers, virtual machines, and so on) from each other.

The MESs attach an MPLS label to each MAC address learned from the CE devices. This label and MAC address combination is advertised to the other MESs in the control plane. Control plane learning enables load balancing and improves convergence times in the event of certain types of network failures. The learning process between the MESs and the CE devices is completed using the method best suited to each CE device (data plane learning, IEEE 802.1, LLDP, 802.1aq, and so on).

The policy attributes of an EVPN are similar to an IP VPN (for example, Layer 3 VPNs). Each EVPN routing instance requires that you configure a route distinguisher and one or more route targets. A CE device attaches to an EVPN routing instance on an MES through an Ethernet interface that might be configured for one or more VLANs.

The following features are available for EVPNs:

- Ethernet connectivity between data centers spanning metropolitan area networks (MANs) and WANs
- One or more VLANs for each MAC VPN
- Automatic route distinguishers
- Dual-homed EVPN connection with active standby multihoming

The following features are not supported for EVPNs:

- Graceful restart, graceful Routing Engine switchover (GRES), and nonstop active routing (NSR)
- Active active multihoming

**Related  
Documentation**

- [Supported EVPN Standards on page 5](#)

## CHAPTER 2

# EVPN Standards

- [Supported EVPN Standards on page 5](#)

### Supported EVPN Standards

---

Junos OS substantially supports the following RFCs and Internet drafts that define standards for EVPNs.

- RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*
- RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*
- RFC 7432, *BGP MPLS-Based Ethernet VPN*

The following features are not supported:

- Auto derivation of Ethernet segment (ES) values. Only static ES configurations are supported.
- Host proxy ARP.
- MAC mobility extended community.
- VLAN bundle service interface.

- |                                  |                                                                                                                                       |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Related<br/>Documentation</b> | <ul style="list-style-type: none"><li>• <i>EVPN Overview</i></li><li>• <i>Accessing Standards Documents on the Internet</i></li></ul> |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|



## CHAPTER 3

# Distributing Routes in VPNs

- [Configuring an IGP on the PE and P Routers on EX9200 Switches on page 7](#)
- [Configuring IBGP Sessions Between PE Routers in VPNs on EX9200 Switches on page 7](#)
- [Configuring a Signaling Protocol and LSPs for VPNs on EX9200 Switches on page 9](#)

### Configuring an IGP on the PE and P Routers on EX9200 Switches

---

For Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, VPLS, EVPNs, and Layer 2 circuits to function properly, the service provider's PE and P routers must be able to exchange routing information. For this to happen, you must configure either an IGP (such as OSPF or IS-IS) or static routes on these routers. You configure the IGP on the master instance of the routing protocol process at the **[edit protocols]** hierarchy level, not within the routing instance used for the VPN—that is, not at the **[edit routing-instances]** hierarchy level.

When you configure the PE router, do not configure any summarization of the PE router's loopback addresses at the area boundary. Each PE router's loopback address should appear as a separate route.

#### Related Documentation

- *Example: Configuring IS-IS*
- *Examples: Configuring Static Routes*
- *OSPF Feature Guide for Routing Devices*

### Configuring IBGP Sessions Between PE Routers in VPNs on EX9200 Switches

---

You must configure an IBGP session between the PE routers to allow the PE routers to exchange information about routes originating and terminating in the VPN. The PE routers rely on this information to determine which labels to use for traffic destined for remote sites.

Configure an IBGP session for the VPN as follows:

```
[edit protocols]
bgp {
  group group-name {
    type internal;
```

```
local-address ip-address;  
family evpn {  
    signaling;  
}  
family (inet-vpn | inet6-vpn) {  
    unicast;  
}  
family l2vpn {  
    signaling;  
}  
neighbor ip-address;  
}
```

The IP address in the **local-address** statement is the address of the loopback interface on the local PE router. The IBGP session for the VPN runs through the loopback address. (You must also configure the loopback interface at the **[edit interfaces]** hierarchy level.)

The IP address in the **neighbor** statement is the loopback address of the neighboring PE router.

The **family** statement allows you to configure the IBGP session for Layer 2 VPNs, VPLS, EVPNs or for Layer 3 VPNs.

- To configure an IBGP session for Layer 2 VPNs and VPLS, include the **signaling** statement at the **[edit protocols bgp group *group-name* family l2vpn]** hierarchy level:

```
[edit protocols bgp group group-name family l2vpn]  
signaling;
```

- To configure an IBGP session for EVPNs, include the **signaling** statement at the **[edit protocols bgp group *group-name* family evpn]** hierarchy level:

```
[edit protocols bgp group group-name family evpn]  
signaling;
```

- To configure an IPv4 IBGP session for Layer 3 VPNs, configure the **unicast** statement at the **[edit protocols bgp group *group-name* family inet-vpn]** hierarchy level:

```
[edit protocols bgp group group-name family inet-vpn]  
unicast;
```

- To configure an IPv6 IBGP session for Layer 3 VPNs, configure the **unicast** statement at the **[edit protocols bgp group *group-name* family inet6-vpn]** hierarchy level:

```
[edit protocols bgp group group-name family inet6-vpn]  
unicast;
```



**NOTE:** You can configure both **family inet** and **family inet-vpn** or both **family inet6** and **family inet6-vpn** within the same peer group. This allows you to enable support for both IPv4 and IPv4 VPN routes or both IPv6 and IPv6 VPN routes within the same peer group.

---

- Related Documentation**
- [Configuring a Signaling Protocol and LSPs for VPNs on EX9200 Switches on page 9](#)

## Configuring a Signaling Protocol and LSPs for VPNs on EX9200 Switches

For VPNs to function, you must enable the LDP signaling protocol on the provider edge (PE) routers and on the provider (P) routers.

To enable the LDP signaling protocol, perform the steps in the following section:

- [Using LDP for VPN Signaling on page 9](#)

### Using LDP for VPN Signaling

To use LDP for VPN signaling, perform the following steps on the PE and provider (P) routers:

1. Configure LDP on the interfaces in the core of the network by including the **ldp** statement at the **[edit protocols]** hierarchy level.

You need to configure LDP only on the interfaces between PE routers or between PE and P routers. You can think of these as the “core-facing” interfaces. You do not need to configure LDP on the interface between the PE and customer edge (CE) routers.

```
[edit]
protocols {
  ldp {
    interface type-fpc/pic/port;
  }
}
```

2. Configure the MPLS address family on the interfaces on which you enabled LDP (the interfaces you configured in Step 1) by including the **family mpls** statement at the **[edit interfaces type-fpc/pic/port unit logical-unit-number]** hierarchy level.

```
[edit]
interfaces {
  type-fpc/pic/port {
    unit logical-unit-number {
      family mpls;
    }
  }
}
```

3. Configure OSPF or IS-IS on each PE and P router.

You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

- To configure OSPF, include the **ospf** statement at the **[edit protocols]** hierarchy level. At a minimum, you must configure a backbone area on at least one of the router's interfaces.

```
[edit]
protocols {
  ospf {
```

```
        area 0.0.0.0 {  
            interface type-fpc/pic/port;  
        }  
    }  
}
```

- To configure IS-IS, include the **isis** statement at the **[edit protocols]** hierarchy level and configure the loopback interface and International Organization for Standardization (ISO) family at the **[edit interfaces]** hierarchy level. At a minimum, you must enable IS-IS on the router, configure a network entity title (NET) on one of the router's interfaces (preferably the loopback interface, lo0), and configure the ISO family on all interfaces on which you want IS-IS to run. When you enable IS-IS, Level 1 and Level 2 are enabled by default. The following is the minimum IS-IS configuration. In the **address** statement, **address** is the NET.

```
[edit]  
interfaces {  
    lo0 {  
        unit logical-unit-number {  
            family iso {  
                address address;  
            }  
        }  
    }  
    type-fpc/pic/port {  
        unit logical-unit-number {  
            family iso;  
        }  
    }  
}  
protocols {  
    isis {  
        interface all;  
    }  
}
```

For more information about configuring OSPF and IS-IS, see the *OSPF Feature Guide for Routing Devices* and *IS-IS Feature Guide for Routing Devices*.

**Related Documentation**

- [EVPN Overview for Switches on page 3](#)



## PART 2

# Configuring EVPNs

- [Configuring Routing Instances for EVPN on page 13](#)
- [Configuring Integrated Bridging and Routing Using EVPNs on page 17](#)



# Configuring Routing Instances for EVPN

- [Configuring EVPN Routing Instances on EX9200 Switches on page 13](#)
- [Tracing EVPN Traffic and Operations on page 15](#)

## Configuring EVPN Routing Instances on EX9200 Switches

---

To configure an EVPN routing instance, complete the following configuration on the PE router (or on the MPLS edge switch) within the EVPN service provider's network:

1. Configure the EVPN routing instance name using the **routing-instances** statement at the **[edit]** hierarchy level:

```
routing-instances routing-instance-name {...}
```

2. Configure the **evpn** option for the **instance-type** statement at the **[edit routing-instances routing-instance-name]** hierarchy level:

```
instance-type evpn;
```

3. Configure the interfaces for handling EVPN traffic between the MES and the CE device using the **interface** statement at the **[edit routing-instances routing-instance-name]** hierarchy level:

```
interface interface-name;
```

4. Configure a VLAN identifier for the EVPN routing instance using the **vlan-id** statement at the **[edit routing-instances routing-instance-name]** hierarchy level:

```
vlan-id (vlan-id | all | none);
```

5. Configure a route distinguisher on a PE router by including the **route-distinguisher** statement:

```
route-distinguisher (as-number:number | ip-address:number);
```

Each routing instance that you configure on a PE router must have a unique route distinguisher associated with it. VPN routing instances need a route distinguisher to help BGP to distinguish between potentially identical network layer reachability information (NLRI) messages received from different VPNs. If you configure different VPN routing instances with the same route distinguisher, the commit fails.

For a list of the hierarchy levels at which you can include this statement, see the statement summary for this statement.

The route distinguisher is a 6-byte value that you can specify in one of the following formats:

- **as-number:number**, where **as-number** is an autonomous system (AS) number (a 2-byte value) and **number** is any 4-byte value. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider's (ISP's) own or the customer's own AS number.
  - **ip-address:number**, where **ip-address** is an IP address (a 4-byte value) and **number** is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the **router-id** statement, which is a nonprivate address in your assigned prefix range.
6. Configure either import and export policies for the EVPN routing table, or configure the default policies using the **vrf-target** statement configured at the **[edit routing-instances routing-instance-name]** hierarchy level.
- See *Configuring Policies for the VRF Table on PE Routers in VPNs*.
7. Configure each EVPN interface for the EVPN routing instance:
- a. Configure interface encapsulation for the CE facing interfaces at the **[edit interfaces interface-name encapsulation]** hierarchy level. . Supported encapsulations for EX9200 switches are: (**extended-vlan-bridge** | **flexible-ethernet-services** | **vlan-bridge**).
  - b. Configure **vlan-bridge** encapsulation on the logical interface at the **[edit interfaces interface-name flexible-vlan-tagging encapsulation flexible-ethernet-services unit 0 encapsulation]** hierarchy level.
  - c. (Optional) Allow the EVPN to establish a connection to the CE device even if the CE device interface encapsulation and the EVPN interface encapsulations do not match by including the **ignore-encapsulation-mismatch** statement at the **[edit routing-instances routing-instance-name protocols evpn interface interface-name]** hierarchy level.
8. Specify the maximum number of media access control (MAC) addresses that can be learned by the EVPN routing instance by including the **interface-mac-limit** statement.

You can configure the same limit for all interfaces configured for a routing instance by including this statement at the **[edit routing-instances routing-instance-name protocols evpn]** hierarchy level. You can also configure a limit for a specific interface by including this statement at the **[edit routing-instances routing-instance-name protocols evpn interface interface-name]** hierarchy level.

By default, packets with new source MAC addresses are forwarded after the MAC address limit is reached. You can alter this behavior by including the **packet-action drop** statement at either the **[edit routing-instances routing-instance-name protocols evpn interface-mac-limit]** or the **[edit routing-instances routing-instance-name protocols evpn interface interface-name]** hierarchy level. If you configure this statement, packets from new source MAC addresses are dropped once the configured MAC address limit is reached.

9. Enable MAC accounting for the EVPN by including the `mac-statistics` statement at the `[edit routing-instances routing-instance-name protocols evpn]` hierarchy level.
10. Specify the number of addresses that can be stored in the MAC routing table using the `mac-table-size` statement at the `[edit routing-instances routing-instance-name protocols evpn]` hierarchy level.

You can optionally configure the `packet-action drop` option to specify that packets for new source MAC addresses be dropped once the MAC address limit is reached. If you do not configure this option, packets for new source MAC addresses are forwarded.

11. Disable MAC learning by including the `no-mac-learning` statement at either the `[edit routing-instances routing-instance-name protocols evpn]` hierarchy level to apply this behavior to all of the devices configured for an EVPN routing instance or at the `[edit routing-instances routing-instance-name protocols evpn interface interface-name]` hierarchy level to apply this behavior to just one of the CE devices.

**Related Documentation**

- *Configuring Policies for the VRF Table on PE Routers in VPNs*
- *Configuring Routing Instances on PE Routers in VPNs*
- [Tracing EVPN Traffic and Operations on page 15](#)

## Tracing EVPN Traffic and Operations

To configure the EVPN routing instance to trace a variety of different parameters related to EVPN operation:

1. Specify the name of one or more EVPN trace files using the `file` option for the `traceoptions` statement at the `[edit routing-instances routing-instance-name protocols evpn]` hierarchy level:

```
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>;
}
```

The `file` option includes the following sub-options:

- **filename**—Specify the 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`.
- **files number**—(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 specified maximum **number** of trace files specified is reached. Then the oldest trace file is overwritten.
- **size size**—(Optional) Maximum size of each trace file. 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.
- **world-readable | no-world-readable**—(Optional) Enable unrestricted file access or restrict file access to the user who created the file.

2. Specify the **flag** option for the **traceoptions** statement:

```
traceoptions {  
  flag flag <flag-modifier> <disable>;  
}
```

The **flag** option allows you to specify the scope of the trace by including one of the following sub-options:

- **all**—All EVPN tracing options
- **error**—Error conditions
- **general**—General events
- **mac-database**—MAC route database in the EVPN routing instance
- **nlri**—EVPN advertisements received or sent by means of the BGP
- **normal**—Normal events
- **oam**—OAM messages
- **policy**—Policy processing
- **route**—Routing information
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing
- **topology**—EVPN topology changes caused by reconfiguration or advertisements received from other PE routers using BGP

You can also specify one of the following modifiers for any of the traceoptions flags:

- **detail**—Provide detailed trace information.
- **disable**—Disable this trace flag.
- **receive**—Trace received packets.
- **send**—Trace sent packets.

- Related Documentation**
- *Configuring EVPN Routing Instances*
  - [traceoptions on page 57](#)

## CHAPTER 5

# Configuring Integrated Bridging and Routing Using EVPNs

- [An EVPN with IRB Solution on EX9200 Switches Overview on page 17](#)
- [Configuring an EVPN with IRB Solution on EX9200 Switches on page 22](#)
- [Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 23](#)

### An EVPN with IRB Solution on EX9200 Switches Overview

A data center service provider (DCSP) hosts the data center for its multiple customers on a common physical network. To each customer (also called a tenant), the service looks like a full-fledged data center that can expand to 4094 VLANs and all private subnets. For disaster recovery, high availability, and optimization of resource utilization, it is common for the DCSP to span the data center over multiple sites. To deploy data center services, a DCSP faces the following main challenges:

- Extending Layer 2 domains across more than one data center site. This requires optimal intra-subnet traffic forwarding.
- Supporting optimal inter-subnet traffic forwarding and optimal routing in the event of virtual machine (VM) motion.
- Supporting multiple tenants with independent VLAN and subnet space.

Ethernet VPN (EVPN) is targeted to handle all the preceding challenges, wherein:

- The basic EVPN functionality enables optimal intra-subnet traffic forwarding
- Implementing the integrated routing and bridging (IRB) solution in an EVPN deployment enables optimal inter-subnet traffic forwarding
- Configuring EVPN with virtual switch support enables multiple tenants with independent VLAN and subnet space

The following sections describe the integrated routing and bridging (IRB) solution for EVPNs:

- [Need for an EVPN IRB Solution on page 18](#)
- [Implementing the EVPN IRB Solution on page 18](#)
- [Benefits of Implementing the EVPN IRB Solution on page 20](#)

## Need for an EVPN IRB Solution

EVPN is a technology used to provide Layer 2 extension and interconnection across an IP/MPLS core network to different physical sites belonging to a single Layer 2 domain. In a data center environment with EVPN, there is a need for both Layer 2 (intra-subnet traffic) and Layer 3 (inter-subnet traffic) forwarding and potentially interoperability with tenant Layer 3 VPNs.

With only a Layer 2 solution, there is no optimum forwarding of inter-subnet traffic, even when the traffic is local, for instance, when both the subnets are on the same server.

With only a Layer 3 solution, the following issues for intra-subnet traffic can arise:

- MAC address aliasing issue where duplicate MAC addresses are not detected.
- TTL issue for applications that use TTL 1 to confine traffic within a subnet.
- IPv6 link-local addressing and duplicate address detection that relies on Layer 2 connectivity.
- Layer 3 forwarding does not support the forwarding semantics of a subnet broadcast.
- Support of non-IP applications that require Layer 2 forwarding.

Because of the above mentioned shortcomings of a pure Layer 2 and Layer 3 solution, there is a need for a solution incorporating optimal forwarding of both Layer 2 and Layer 3 traffic in the data center environment when faced with operational considerations such as Layer 3 VPN interoperability and virtual machine (VM) mobility.

An EVPN-based integrated routing and bridging (IRB) solution provides optimum unicast and multicast forwarding for both intra-subnets and inter-subnets within and across data centers.

The EVPN IRB feature is useful for service providers operating in an IP/MPLS network that provides both Layer 2 VPN or VPLS services and Layer 3 VPN services who want to extend their service to provide cloud computation and storage services to their existing customers.

## Implementing the EVPN IRB Solution

An EVPN IRB solution provides the following:

- Optimal forwarding for intra-subnet (Layer 2) traffic.
- Optimal forwarding for inter-subnet (Layer 3) traffic.
- Support for ingress replication for multicast traffic.
- Support for network-based as well as host-based overlay models.
- Support for consistent policy-based forwarding for both Layer 2 and Layer 3 traffic.

Junos OS supports several models of EVPN configuration to satisfy the individual needs of EVPN and data center cloud services customers. To provide flexibility and scalability, multiple VLANs can be defined within a particular EVPN instance. Likewise, one or more



EVPN instances can be associated with a single Layer 3 VPN virtual routing and forwarding (VRF). In general, each data center tenant is assigned a unique Layer 3 VPN VRF, while a tenant could comprise one or more EVPN instances and one or more VLANs per EVPN instance. To support this model, each configured VLAN (including the default VLAN for an EVPN instance) requires an IRB interface to perform the Layer 2 and Layer 3 functions. Each VLAN or IRB interface maps to a unique IP subnet in the VRF.

There are two major functions that are supported for IRB in EVPN.

- Host MAC-IP synchronization

This includes:

- Advertising the IP address along with the MAC advertisement route in EVPN. This is done by using the IP field in the EVPN MAC advertisement route.
- The receiving PE router installs MAC into the EVPN instance (EVI) table and installs IP into the associated VRF.

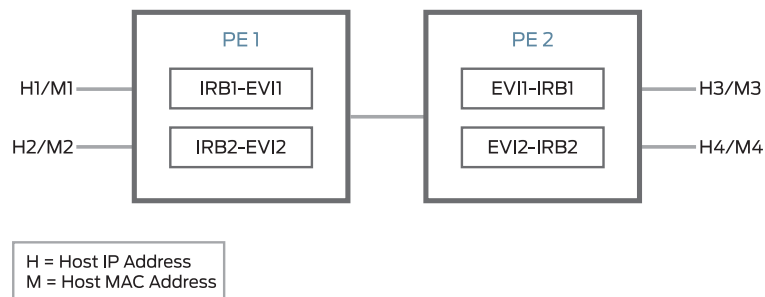
- Gateway MAC-IP synchronization

This includes:

- Advertising all local IRB MAC and IP addresses in an EVPN. This is achieved by including the default gateway extended community in the EVPN MAC advertisement route.
- The receiving PE creates a forwarding state to route packets destined for the gateway MAC, and a proxy ARP is done for the gateway IP with the MAC advertised in the route.

Figure 1 on page 19 illustrates the inter-subnet traffic forwarding between two provider edge (PE) devices—PE1 and PE2. The IRB1 and IRB2 interfaces on each PE device belong to a different subnet, but they share a common VRF.

**Figure 1: Inter-Subnet Traffic Forwarding**



The inter-subnet traffic forwarding is performed as follows:

1. PE2 advertises the H3-M3 and H4-M4 binding to PE1. Similarly, PE1 advertises the H1-M1 and H2-M2 binding to PE2.
2. PE1 and PE2 install the MAC address in the corresponding EVI MAC table, whereas the IP routes are installed in the shared VRF.

3. The advertising PE device is set as the next hop for the IP routes.
4. If H1 sends packets to H4, the packets are sent to IRB1 on PE1.
5. IP lookup for H4 happens in the shared VRF on PE1. Because the next hop for the H4 IP is PE2 (the advertising PE), an IP unicast packet is sent to PE2.
6. PE1 rewrites the MAC header based on the information in the VRF route, and PE2 performs a MAC lookup to forward the packet to H4.

## Benefits of Implementing the EVPN IRB Solution

The main goal of the EVPN IRB solution is to provide optimal Layer 2 and Layer 3 forwarding. The solution is required to efficiently handle inter-subnet forwarding as well as virtual machine (VM) mobility. VM mobility refers to the ability of a VM to migrate from one server to another within the same or a different data center while retaining its existing MAC and IP address. Providing optimal forwarding for inter-subnet traffic and effective VM mobility involves solving two problems – the default gateway problem and the triangular routing problem.

- [Gateway MAC and IP Synchronization on page 20](#)
- [Layer 3 VPN Interworking on page 20](#)

### Gateway MAC and IP Synchronization

---

In an EVPN IRB deployment, the IP default gateway for a VM is the IP address configured on the IRB interface of the provider edge (PE) router corresponding to the VLAN of which the VM is a member. The default gateway problem arises because a VM does not flush its ARP table when relocating from one server to another and continues sending packets with the destination MAC address set to that of the original gateway. If the old and new servers are not part of the same Layer 2 domain (the new Layer 2 domain could be within the current data center or a new data center), the gateway previously identified is no longer the optimal or local gateway. The new gateway needs to identify packets containing the MAC addresses of other gateways on remote PE routers and forward the traffic as if the packets were destined to the local gateway itself. At minimum, this functionality requires each PE router to advertise its gateway or IRB MAC and IP addresses to all other PE routers in the network. The gateway address exchange can be accomplished using the standard MAC route advertisement message (including the IP address parameter) and tagging that route with the default gateway extended community so that the remote PE routers can distinguish the gateway MAC advertisement routes from normal MAC advertisement routes.

### Layer 3 VPN Interworking

---

The inter-data center aspect of the EVPN IRB solution involves routing between VMs that are present in different data centers or routing between a host site completely outside of the data center environment and a VM within a data center. This solution relies on the ability of EVPN MAC route advertisements to carry both MAC address and IP address information. The local MAC learning functionality of the PE router is extended to also capture IP address information associated with MAC addresses learned locally. That IP-MAC address mapping information is then distributed to each PE router through normal EVPN procedures. When a PE router receives such MAC and IP information, it

installs the MAC route in the EVPN instance as well as a host route for the associated IP address in the Layer 3 VPN VRF corresponding to that EVPN instance. When a VM moves from one data center to another, normal EVPN procedures result in the MAC and IP address being advertised from the new PE router which the VM resides behind. The host route installed in the VRF associated with an EVPN solicits Layer 3 traffic destined to that VM to the new PE router and avoids triangular routing between the source, the former PE router the VM resided behind, and the new PE router.

BGP scalability is a potential concern with the inter-data center triangular routing avoidance solution because of the potential for injection of many host routes into Layer 3 VPN. With the method previously described, in the worst case there is an IP host route for each MAC address learned through the local EVPN MAC learning procedures or through a MAC advertisement message received from a remote PE router. BGP route target filtering can be used to limit distribution of such routes.

The following functional elements are required to implement the inter-data center triangular routing avoidance using Layer 3 inter-subnet forwarding procedures:

1. The source host sends an IP packet using its own source MAC and IP address with the destination MAC of the IRB interface of the local PE router and the IP address of the destination host.
2. When the IRB interface receives the frame with its MAC as the destination, it performs a Layer 3 lookup in the VRF associated with the EVPN instance to determine where to route the packet.
3. In the VRF, the PE router finds the Layer 3 route derived from a MAC plus an IP EVPN route received from the remote PE router earlier. The destination MAC address is then changed to the destination MAC address corresponding to the destination IP.
4. The packet is then forwarded to the remote PE router serving the destination host using MPLS, using the label corresponding to the EVPN instance of which the destination host is a member.
5. The egress PE router receiving the packet performs a Layer 2 lookup for the destination host's MAC and sends the packet to the destination host on the attached subnet via the egress PE router's IRB interface.
6. Because the ingress PE router is performing Layer 3 routing, the IP TTL is decremented.

**Related  
Documentation**

- [Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 23](#)
- [EVPN Overview for Switches on page 3](#)

## Configuring an EVPN with IRB Solution on EX9200 Switches

---

You can configure an Ethernet VPN (EVPN) with IRB solution to enable Layer 2 switching and Layer 3 routing operations within a single node, thus avoiding extra hops for inter-subnet traffic. The EVPN IRB solution eliminates the default gateway problem using the gateway MAC and IP synchronization, and avoids the triangular routing problem with Layer 3 interworking by creating IP host routes for virtual machines (VMs) in the tenant virtual routing and forwarding (VRF) routing instances.

Before you begin:

1. Configure the switch interfaces.
2. Configure the router ID and autonomous system number for the device.
3. Enable the chained composite next hop for EVPN.
4. Configure OSPF or any other IGP protocol.
5. Configure a BGP internal group.
6. Include the EVPN signaling network layer reachability information (NLRI) to the internal BGP group.
7. Configure LDP.
8. Configure MPLS.

To configure the PE device:

1. Configure the EVPN routing instance.  

```
[edit routing-instances]
user@PE1# set evpn-instance instance-type evpn
```
2. Set the VLAN identifier for the bridging domain in the EVPN routing instance.  

```
[edit routing-instances]
user@PE1# set evpn-instance vlan-id VLAN-ID
```
3. Configure the interface name for the EVPN routing instance.  

```
[edit routing-instances]
user@PE1# set evpn-instance interface CE-facing-interface
```
4. Configure the IRB interface as the routing interface for the EVPN routing instance.  

```
[edit routing-instances]
user@PE1# set evpn-instance l3-interface irb.0
```
5. Configure the route distinguisher for the EVPN routing instance.  

```
[edit routing-instances]
user@PE1# set evpn-instance route-distinguisher route-distinguisher-value
```
6. Configure the VPN routing and forwarding (VRF) target community for the EVPN routing instance.  

```
[edit routing-instances]
user@PE1# set evpn-instance vrf-target vrf-target-value
```

7. Assign the interface name that connects the PE device site to the VPN.

```
[edit routing-instances]
user@PE1# set evpn-instance protocols evpn interface CE-facing-interface
```

8. Configure the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance instance-type vrf
```

9. Configure the IRB interface as the routing interface for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance interface irb.0
```

10. Configure the route distinguisher for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance route-distinguisher route-distinguisher-value
```

11. Configure the VRF label for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance vrf-table-label
```

12. Verify and commit the configuration.

For example:

```
[edit routing-instances]
user@PE1# set evpna instance-type evpn
user@PE1# set evpna vlan-id 10
user@PE1# set evpna interface ge-1/1/8.0
user@PE1# set evpna l3-interface irb.0
user@PE1# set evpna route-distinguisher 100.255.0.1:100
user@PE1# set evpna vrf-target target:100:100
user@PE1# set evpna protocols evpn interface ge-1/1/8.0
user@PE1# set vrf instance-type vrf
user@PE1# set vrf interface irb.0
user@PE1# set vrf route-distinguisher 100.255.0.1:300
user@PE1# set vrf vrf-target target:100:300
user@PE1# set vrf vrf-table-label

[edit]
user@PE1# commit
commit complete
```

**Related Documentation** • [Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 23](#)

## Example: Configuring an EVPN with IRB Solution on EX9200 Switches

This example shows how to configure an integrated routing and bridging (IRB) solution in an Ethernet VPN (EVPN) deployment.

- [Requirements on page 24](#)
- [Overview on page 24](#)

- [Configuration on page 24](#)
- [Verification on page 30](#)

## Requirements

This example uses the following hardware and software components:

- Two EX9200 switches configured as PE routers
- Junos OS Release 14.2 or later running on all the PE routers

Before you begin:

1. Configure the switch interfaces.
2. Configure OSPF or any other IGP protocol.
3. Configure BGP.
4. Configure LDP.
5. Configure MPLS.

## Overview

In an EVPN solution, multiple VLANs can be defined within a particular EVPN instance, and one or more EVPN instances can be associated with a single Layer 3 VPN VRF. In general, each data center tenant is assigned a unique Layer 3 VPN virtual route forwarding (VRF), although the tenant can comprise one or more EVPN instances or VLANs per EVPN instance.

To support this flexibility and scalability factor, the EVPN solution provides support for the IRB interfaces on EX9200 switches to facilitate optimal Layer 2 and Layer 3 forwarding along with virtual machine mobility. The IRB interfaces are configured on each configured VLAN including the default VLAN for an EVPN instance.

IRB is the ability to do Layer 2 switching and Layer 3 routing within a single node, thus avoiding extra hops for inter-subnet traffic. The EVPN IRB solution eliminates the default gateway problem using the gateway MAC and IP synchronization, and avoids the triangular routing problem with Layer 3 interworking by creating IP host routes for virtual machines (VMs) in the tenant VRFs.

## Configuration

<b>CLI Quick Configuration</b>	To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the <b>[edit]</b> hierarchy level.
<b>CE1</b>	<pre>set interfaces ge-1/1/7 vlan-tagging set interfaces ge-1/1/7 unit 0 vlan-id 10 set interfaces ge-1/1/7 unit 0 family inet address 10.0.0.1/24 set routing-options static route 20.0.0.0/24 next-hop 10.0.0.251</pre>
<b>PE1</b>	<pre>set interfaces ge-1/0/8 unit 0 family inet address 30.0.0.1/30</pre>

```

set interfaces ge-1/0/8 unit 0 family mpls
set interfaces ge-1/1/8 flexible-vlan-tagging
set interfaces ge-1/1/8 encapsulation flexible-ethernet-services
set interfaces ge-1/1/8 unit 0 encapsulation vlan-bridge
set interfaces ge-1/1/8 unit 0 vlan-id 10
set interfaces irb unit 0 family inet address 10.0.0.251/24
set interfaces lo0 unit 0 family inet address 100.255.0.1/32
set routing-options router-id 100.255.0.1
set routing-options autonomous-system 100
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.0.1
set protocols bgp group ibgp family evpn signaling
set protocols bgp group ibgp neighbor 10.255.0.2
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set routing-instances evpna instance-type evpn
set routing-instances evpna vlan-id 10
set routing-instances evpna interface ge-1/1/8.0
set routing-instances evpna l3-interface irb.0
set routing-instances evpna route-distinguisher 100.255.0.1:100
set routing-instances evpna vrf-target target:100:100
set routing-instances evpna protocols evpn interface ge-1/1/8.0
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.0
set routing-instances vrf route-distinguisher 100.255.0.1:300
set routing-instances vrf vrf-target target:100:300
set routing-instances vrf vrf-table-label

```

```

PE2 set interfaces ge-2/0/8 unit 0 family inet address 30.0.0.2/30
set interfaces ge-2/0/8 unit 0 family mpls
set interfaces ge-2/1/8 flexible-vlan-tagging
set interfaces ge-2/1/8 encapsulation flexible-ethernet-services
set interfaces ge-2/1/8 unit 0 encapsulation vlan-bridge
set interfaces ge-2/1/8 unit 0 vlan-id 20
set interfaces irb unit 0 family inet address 20.0.0.251/24
set interfaces lo0 unit 0 family inet address 100.255.0.2/32
set routing-options router-id 100.255.0.2
set routing-options autonomous-system 100
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 100.255.0.2
set protocols bgp group ibgp family evpn signaling
set protocols bgp group ibgp neighbor 100.255.0.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set routing-instances evpna instance-type evpn
set routing-instances evpna vlan-id 20

```

```

set routing-instances evpna interface ge-2/1/8.0
set routing-instances evpna l3-interface irb.0
set routing-instances evpna route-distinguisher 100.255.0.2:100
set routing-instances evpna vrf-target target:200:100
set routing-instances evpna protocols evpn interface ge-2/1/8.0
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.0
set routing-instances vrf route-distinguisher 100.255.0.2:300
set routing-instances vrf vrf-target target:200:300
set routing-instances vrf vrf-table-label

```

```

CE2    set interfaces ge-2/1/7 unit 0 vlan-id 20
       set interfaces ge-2/1/7 unit 0 family inet address 20.0.0.2/24
       set routing-options static route 10.0.0.0/24 next-hop 20.0.0.251

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure Router PE1:



**NOTE:** Repeat this procedure for Router PE2, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure Router PE1 interfaces.

**[edit interfaces]**

```

user@PE1# set ge-1/0/8 unit 0 family inet address 30.0.0.1/30
user@PE1# set ge-1/0/8 unit 0 family mpls

```

```

user@PE1# set ge-1/1/8 flexible-vlan-tagging
user@PE1# set ge-1/1/8 encapsulation flexible-ethernet-services
user@PE1# set ge-1/1/8 unit 0 encapsulation vlan-bridge
user@PE1# set ge-1/1/8 unit 0 vlan-id 10

```

```

user@PE1# set irb unit 0 family inet address 10.0.0.251/24

```

```

user@PE1# set lo0 unit 0 family inet address 100.255.0.1/32

```

2. Set the router ID and autonomous system number for Router PE1.

**[edit routing-options]**

```

user@PE1# set router-id 100.255.0.1
user@PE1# set autonomous-system 100

```

3. Configure the chained composite next hop for EVPN.

**[edit routing-options]**

```

user@PE1# set forwarding-table chained-composite-next-hop ingress evpn

```

4. Enable LDP on all interfaces of Router PE1, excluding the management interface.

**[edit protocols]**

```

user@PE1# set ldp interface all

```



```
user@PE1# set ldp interface fxp0.0 disable
```

5. Enable MPLS on all the interfaces of Router PE1, excluding the management interface.

```
[edit protocols]
user@PE1# set mpls interface all
user@PE1# set mpls mpls interface fxp0.0 disable
```

6. Configure the BGP group for Router PE1.

```
[edit protocols]
user@PE1# set bgp group ibgp type internal
```

7. Assign local and neighbor addresses to the ibgp BGP group for Router PE1 to peer with Router PE2.

```
[edit protocols]
user@PE1# set bgp group ibgp local-address 10.255.0.1
user@PE1# set bgp group ibgp neighbor 10.255.0.2
```

8. Include the EVPN signaling Network Layer Reachability Information (NLRI) to the ibgp BGP group.

```
[edit protocols]
user@PE1# set bgp group ibgp family evpn signaling
```

9. Configure OSPF on all the interfaces of Router PE1, excluding the management interface.

```
[edit protocols]
user@PE1# set ospf area 0.0.0.0 interface all
user@PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
```

10. Configure the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpna instance-type evpn
```

11. Set the VLAN identifier for the bridging domain in the evpna routing instance.

```
[edit routing-instances]
user@PE1# set evpna vlan-id 10
```

12. Configure the interface name for the evpna routing instance.

```
[edit routing-instances]
user@PE1# set evpna interface ge-1/1/8.0
```

13. Configure the IRB interface as the routing interface for the evpna routing instance.

```
[edit routing-instances]
user@PE1# set evpna l3-interface irb.0
```

14. Configure the route distinguisher for the evpna routing instance.

```
[edit routing-instances]
user@PE1# set evpna route-distinguisher 100.255.0.1:100
```

15. Configure the VPN routing and forwarding (VRF) target community for the evpna routing instance.

```
[edit routing-instances]
```

```
user@PE1# set evpna vrf-target target:100:100
```

16. Assign the interface name that connects the PE1 site to the VPN.

```
[edit routing-instances]
user@PE1# set evpna protocols evpn interface ge-1/1/8.0
```

17. Configure the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf instance-type vrf
```

18. Configure the IRB interface as the routing interface for the vrf routing instance.

```
[edit routing-instances]
user@PE1# set vrf interface irb.0
```

19. Configure the route distinguisher for the vrf routing instance.

```
[edit routing-instances]
user@PE1# set vrf route-distinguisher 100.255.0.1:300
```

20. Configure the VRF label for the vrf routing instance.

```
[edit routing-instances]
user@PE1# set vrf vrf-table-label
```

---

## Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, **show protocols**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
ge-1/0/8 {
  unit 0 {
    family inet {
      address 30.0.0.1/30;
    }
    family mpls;
  }
}
ge-1/1/8 {
  flexible-vlan-tagging;
  encapsulation flexible-ethernet-services;
  unit 0 {
    encapsulation vlan-bridge;
    vlan-id 10;
  }
}
irb {
  unit 0 {
    family inet {
      address 10.0.0.251/24;
    }
  }
}
lo0 {
```

```
unit 0 {
  family inet {
    address 100.255.0.1/32 {
    }
  }
}

user@PE1# show routing-options
router-id 10.255.0.1;
autonomous-system 100;
forwarding-table {
  chained-composite-next-hop {
    ingress {
      evpn;
    }
  }
}

user@PE1# show protocols
ldp {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
mpls {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
bgp {
  group ibgp {
    type internal;
    local-address 10.255.0.1;
    family evpn {
      signaling;
    }
    neighbor 10.255.0.2;
  }
}
ospf {
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}

user@PE1# show routing-instances
evpna {
  instance-type evpn;
  vlan-id 10;
  interface ge-1/1/8.0;
  l3-interface irb.0;
```

```
route-distinguisher 100.255.0.1:100;
vrf-target target:100:100;
protocols {
    evpn {
        interface ge-1/1/8.0;
    }
}
vrf {
    instance-type vrf;
    interface irb.0;
    route-distinguisher 100.255.0.1:300;
    vrf-target target:100:300;
    vrf-table-label;
}
```

## Verification

Confirm that the configuration is working properly.

- [Verifying Local IRB MACs on page 30](#)
- [Verifying Remote IRB MACs on page 31](#)
- [Verifying Local IRB IPs on page 32](#)
- [Verifying Remote IRB IPs on page 33](#)
- [Verifying CE-CE Inter-Subnet Forwarding on page 34](#)

### Verifying Local IRB MACs

---

**Purpose** Verify that the local IRB MACs are learned from L2ALD.

**Action** On Router PE1, determine the MAC address of the local IRB interface.

From operational mode, run the **show interfaces irb extensive | match "Current address"** command.

```
user@PE1> show interfaces irb extensive | match "Current address"
Current address: a8:d0:e5:54:0d:10, Hardware address: a8:d0:e5:54:0d:10
```

From operational mode, run the **show route table evpn.0 extensive | find "a8:d0:e5:54:0d:10"** command.

```
user@PE1> show route table evpn.0 extensive | find "a8:d0:e5:54:0d:10"
2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10/384 (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group PE type Internal) Type 1 val 0x2736568 (adv_entry)
  Advertised metrics:
    Flags: Nexthop Change
    Nexthop: Self
    Localpref: 100
    AS path: [100] I
    Communities: target:100:100 evpn-default-gateway
Path 2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10 Vector len 4. Val: 0
  *EVPN Preference: 170
    Next hop type: Indirect
```

```

Address: 0x26f8354
Next-hop reference count: 6
Protocol next hop: 10.255.0.1
Indirect next hop: 0x0 - INH Session ID: 0x0
State: <Active Int Ext>
Age: 23:29:08
Validation State: unverified
Task: evpna-evpn
Announcement bits (1): 1-BGP_RT_Background
AS path: I
Communities: evpn-default-gateway
Route Label: 299776

```

**Meaning** The MAC-only route for the local IRB interface appears in the EVPN instance route table on Router PE1 and is learned from EVPN and tagged with the default gateway extended community.

### Verifying Remote IRB MACs

**Purpose** Verify that the remote IRB MACs are learned from BGP.

**Action** On Router PE1, determine the MAC address of the local IRB interface.

From operational mode, run the **show interfaces irb extensive | match "Current address"** command.

```

user@PE1> show interfaces irb extensive | match "Current address"
Current address: a8:d0:e5:54:0d:10, Hardware address: a8:d0:e5:54:0d:10

```

On Router PE2, verify that the remote IRB MACs are learned.

From operational mode, run the **show route table evpna.evpn.0 extensive | find "a8:d0:e5:54:0d:10"** command.

```

user@PE2> show route table evpna.evpn.0 extensive | find "a8:d0:e5:54:0d:10"
2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10/384 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 2.91.223.24:100
    Next hop type: Indirect
    Address: 0x26f8d6c
    Next-hop reference count: 10
    Source: 10.255.0.1
    Protocol next hop: 10.255.0.1
    Indirect next hop: 0x2 no-forward INH Session ID: 0x0
    State: <Secondary Active Int Ext>
    Local AS: 100 Peer AS: 100
    Age: 23:22:17 Metric2: 1
    Validation State: unverified
    Task: BGP_100.10.255.0.1
    Announcement bits (1): 0-evpna-evpn
    AS path: I
    Communities: target:100:100 evpn-default-gateway
    Import Accepted
    Route Label: 299776
    Localpref: 100
    Router ID: 10.255.0.1
    Primary Routing Table bgp.evpn.0
    Indirect next hops: 1

```

```

Protocol next hop: 10.255.0.1 Metric: 1
Indirect next hop: 0x2 no-forward INH Session ID: 0x0
Indirect path forwarding next hops: 1
  Next hop type: Router
  Next hop: 1.0.0.1 via ge-1/0/8.0
  Session Id: 0x1
10.255.0.1/32 Originating RIB: inet.3
  Metric: 1                               Node path count: 1
  Forwarding nexthops: 1
  Nexthop: 1.0.0.1 via ge-1/0/8.0

```

**Meaning** The MAC-only route for the remote IRB interface appears in the EVPN instance route table on Router PE2 and is learned from BGP and tagged with the default gateway extended community.

### Verifying Local IRB IPs

**Purpose** Verify that the local IRB IPs are learned locally by RPD.

**Action** On Router PE1, determine the MAC and IP addresses of the local IRB interface.

From operational mode, run the **show interfaces irb extensive | match "Current address"** command.

```

user@PE1> show interfaces irb extensive | match "Current address"
Current address: a8:d0:e5:54:0d:10, Hardware address: a8:d0:e5:54:0d:10

```

From operational mode, run the **show interfaces irb.0 terse | match inet** command.

```

user@PE1> show interfaces irb.0 terse | match inet
irb.0                up    up    inet    10.0.0.251/24

```

From operational mode, run the **show route table evpn.0 extensive | find "a8:d0:e5:54:0d:10::10.0.0.251"** command.

```

user@PE2> show route table evpn.0 extensive | find "a8:d0:e5:54:0d:10::10.0.0.251"
2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10::10.0.0.251/384 (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group PE type Internal) Type 1 val 0x27365a0 (adv_entry)
  Advertised metrics:
    Flags: Nexthop Change
    Nexthop: Self
    Localpref: 100
    AS path: [100] I
    Communities: target:100:100 evpn-default-gateway
Path 2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10::10.0.0.251 Vector len 4. Val:
0
    *EVPN Preference: 170 <<<<<
      Next hop type: Indirect
      Address: 0x26f8354
      Next-hop reference count: 6
      Protocol next hop: 10.255.0.1
      Indirect next hop: 0x0 - INH Session ID: 0x0
      State: <Active Int Ext>
      Age: 23:48:46
      Validation State: unverified
      Task: evpn-evpn

```

```

Announcement bits (1): 1-BGP_RT_Background
AS path: I
Communities: evpn-default-gateway
Route Label: 299776

```

**Meaning** The MAC plus IP route for the local IRB interface appears in the EVPN instance route table on Router PE1 and is learned from EVPN and tagged with the default gateway extended community.

### Verifying Remote IRB IPs

**Purpose** Verify that the remote IRB IPs are learned from BGP.

**Action** On Router PE1, determine the MAC and IP addresses of the local IRB interface.

From operational mode, run the **show interfaces irb extensive | match "Current address"** command.

```

user@PE1> show interfaces irb extensive | match "Current address"
Current address: a8:d0:e5:54:0d:10, Hardware address: a8:d0:e5:54:0d:10

```

From operational mode, run the **show interfaces irb.0 terse | match inet** command.

```

user@PE1> show interfaces irb.0 terse | match inet
irb.0                up    up    inet    10.0.0.251/24

```

On Router PE2, verify that the remote IRB IPs are learnt.

From operational mode, run the **show route table evpn.evpn.0 extensive | find "a8:d0:e5:54:0d:10::10.0.0.251"** command.

```

user@PE2> show route table evpn.evpn.0 extensive | find "a8:d0:e5:54:0d:10::10.0.0.251"
2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10::10.0.0.251/384 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 2.91.223.216:100
    Next hop type: Indirect
    Address: 0x26f8d6c
    Next-hop reference count: 10
    Source: 10.255.0.1
    Protocol next hop: 10.255.0.1
    Indirect next hop: 0x2 no-forward INH Session ID: 0x0
    State: <Secondary Active Int Ext>
    Local AS: 100 Peer AS: 100
    Age: 23:56:36 Metric2: 1
    Validation State: unverified
    Task: BGP_100.10.255.0.1
    Announcement bits (1): 0-evpn-evpn
    AS path: I
    Communities: target:100:100 evpn-default-gateway
    Import Accepted
    Route Label: 299776
    Localpref: 100
    Router ID: 10.255.0.1
    Primary Routing Table bgp.evpn.0
    Indirect next hops: 1
      Protocol next hop: 10.255.0.1 Metric: 1
      Indirect next hop: 0x2 no-forward INH Session ID: 0x0
      Indirect path forwarding next hops: 1

```

```

Next hop type: Router
Next hop: 1.0.0.1 via ge-1/0/8.0
Session Id: 0x1
10.255.0.1/32 Originating RIB: inet.3
Metric: 1 Node path count: 1
Forwarding nexthops: 1
Nexthop: 1.0.0.1 via ge-1/0/8.0

```

**Meaning** The MAC plus IP route for the remote IRB interface appears in the EVPN instance route table on Router PE2 and is tagged with the default gateway extended community.

### Verifying CE-CE Inter-Subnet Forwarding

**Purpose** Verify inter-subnet forwarding between Routers CE1 and CE2.

**Action** From operational mode, run the **show route table inet.0** command.

```

user@CE1> show route table inet.0
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:15:09
                  > to 10.0.0.251 via ge-1/1/7.0
10.0.0.0/24        *[Direct/0] 1d 23:24:30
                  > via ge-1/1/7.0
10.0.0.1/32        *[Local/0] 1d 23:24:38
                  Local via ge-1/1/7.0

```

From operational mode, run the **ping** command.

```

user@CE1> ping 20.0.0.2 interval 0.1 count 10
PING 20.0.0.2 (20.0.0.2): 56 data bytes
64 bytes from 20.0.0.2: icmp_seq=0 ttl=63 time=0.919 ms
64 bytes from 20.0.0.2: icmp_seq=1 ttl=63 time=0.727 ms
64 bytes from 20.0.0.2: icmp_seq=2 ttl=63 time=0.671 ms
64 bytes from 20.0.0.2: icmp_seq=3 ttl=63 time=0.671 ms
64 bytes from 20.0.0.2: icmp_seq=4 ttl=63 time=0.666 ms
64 bytes from 20.0.0.2: icmp_seq=5 ttl=63 time=0.704 ms
64 bytes from 20.0.0.2: icmp_seq=6 ttl=63 time=0.763 ms
64 bytes from 20.0.0.2: icmp_seq=7 ttl=63 time=0.750 ms
64 bytes from 20.0.0.2: icmp_seq=8 ttl=63 time=12.967 ms
64 bytes from 20.0.0.2: icmp_seq=9 ttl=63 time=0.752 ms

--- 20.0.0.2 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.666/1.959/12.967/3.670 ms

```

**Meaning** Ping from Router CE1 to Router CE2 is successful.

**Related Documentation**

- [An EVPN with IRB Solution on EX9200 Switches Overview on page 17](#)



## PART 3

# Configuration Statements and Operational Commands

- Configuration Statements on page 37
- Operational Commands on page 61



## CHAPTER 6

# Configuration Statements

- [encapsulation \(Physical Interface\) on page 38](#)
- [evpn on page 43](#)
- [extended-vlan-list on page 44](#)
- [instance-type on page 45](#)
- [interface \(EVPN Routing Instances\) on page 47](#)
- [interface \(Routing Instances\) on page 48](#)
- [interface-mac-limit \(VPLS\) on page 49](#)
- [mac-statistics on page 50](#)
- [mac-table-size on page 51](#)
- [no-mac-learning on page 52](#)
- [packet-action on page 54](#)
- [routing-instances on page 56](#)
- [traceoptions \(Protocols EVPN\) on page 57](#)
- [vlan-id \(routing instance\) on page 59](#)

## encapsulation (Physical Interface)

<b>Syntax</b>	encapsulation (atm-ccc-cell-relay   atm-pvc   cisco-hdlc   cisco-hdlc-ccc   cisco-hdlc-tcc   ethernet-bridge   ethernet-ccc   ethernet-over-atm   ethernet-tcc   ethernet-vpls   ethernet-vpls-fr   ether-vpls-over-atm-llc   ethernet-vpls-ppp   extended-frame-relay-ccc   extended-frame-relay-ether-type-tcc   extended-frame-relay-tcc   extended-vlan-bridge   extended-vlan-ccc   extended-vlan-tcc   extended-vlan-vpls   flexible-ethernet-services   flexible-frame-relay   frame-relay   frame-relay-ccc   frame-relay-ether-type   frame-relay-ether-type-tcc   frame-relay-port-ccc   frame-relay-tcc   generic-services   multilink-frame-relay-uni-nni   ppp   ppp-ccc   ppp-tcc   vlan-ccc   vlan-vci-ccc   vlan-vpls);
<b>Hierarchy Level</b>	[edit interfaces <i>interface-name</i> ], [edit interfaces rlsq <i>number:number</i> ]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 11.1 for EX Series switches. Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers ( <b>flexible-ethernet-services</b> , <b>ethernet-ccc</b> , and <b>ethernet-tcc</b> options only).
<b>Description</b>	Specify the physical link-layer encapsulation type. Not all encapsulation types are supported on the switches. See the switch CLI.
<b>Default</b>	<b>ppp</b> —Use serial PPP encapsulation.
<b>Options</b>	<p><b>atm-ccc-cell-relay</b>—Use ATM cell-relay encapsulation.</p> <p><b>atm-pvc</b>—Use ATM PVC encapsulation.</p> <p><b>cisco-hdlc</b>—Use Cisco-compatible High-Level Data Link Control (HDLC) framing.</p> <p><b>cisco-hdlc-ccc</b>—Use Cisco-compatible HDLC framing on CCC circuits.</p> <p><b>cisco-hdlc-tcc</b>—Use Cisco-compatible HDLC framing on TCC circuits for connecting different media.</p> <p><b>ethernet-bridge</b>—Use Ethernet bridge encapsulation on Ethernet interfaces that have bridging enabled and that must accept all packets.</p> <p><b>ethernet-ccc</b>—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, CCC is not supported.</p> <p><b>ethernet-over-atm</b>—For interfaces that carry IPv4 traffic, use Ethernet over ATM encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces. As defined in RFC 2684, <i>Multiprotocol Encapsulation over ATM Adaptation Layer 5</i>, this encapsulation type allows ATM interfaces to connect to devices that support only bridge protocol data units (BPDUs). Junos OS does not completely support bridging, but accepts BPDUs packets as a default gateway. If you use the router as an edge device, then the router acts as a default gateway. It accepts Ethernet LLC/SNAP frames with IP or ARP in the payload, and drops the rest. For packets destined to the Ethernet LAN, a route lookup is done using the destination</p>

IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and MAC header, and the packet is forwarded to the ATM interface.

**ethernet-tcc**—For interfaces that carry IPv4 traffic, use Ethernet TCC encapsulation on interfaces that must accept packets carrying standard TPID values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, TCC is not supported.

**ethernet-vpls**—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard TPID values. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

**ethernet-vpls-fr**—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer Layer 2 Frame Relay connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use the MAC address to forward the packet into a given VPLS instance.

**ethernet-vpls-ppp**—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer Layer 2 PPP connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.

**ether-vpls-over-atm-llc**—For ATM intelligent queuing (IQ) interfaces only, use the Ethernet virtual private LAN service (VPLS) over ATM LLC encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3 encapsulated Ethernet frames with the frame check sequence (FCS) field removed.

**extended-frame-relay-ccc**—Use Frame Relay encapsulation on CCC circuits. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to CCC.

**extended-frame-relay-ether-type-tcc**—Use extended Frame Relay ether type TCC for Cisco-compatible Frame Relay for DLCIs 1 through 1022. This encapsulation type is used for circuits with different media on either side of the connection.

**extended-frame-relay-tcc**—Use Frame Relay encapsulation on TCC circuits to connect different media. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to TCC.

**extended-vlan-bridge**—Use extended VLAN bridge encapsulation on Ethernet interfaces that have IEEE 802.1Q VLAN tagging and bridging enabled and that must accept packets carrying TPID 0x8100 or a user-defined TPID.

**extended-vlan-ccc**—Use extended VLAN encapsulation on CCC circuits with Gigabit Ethernet and 4-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, extended VLAN CCC is not supported. For 4-port Gigabit Ethernet PICs, extended VLAN CCC is not supported.

**extended-vlan-tcc**—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. For 4-port Gigabit Ethernet PICs, extended VLAN TCC is not supported.

**extended-vlan-vpls**—Use extended VLAN VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.



**NOTE:** The built-in Gigabit Ethernet PIC on an M7i router does not support extended VLAN VPLS encapsulation.

---

**flexible-ethernet-services**—For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. Aggregated Ethernet bundles can use this encapsulation type. This encapsulation type allows you to configure any combination of route, TCC, CCC, Layer 2 virtual private networks (VPNs), and VPLS encapsulations on a single physical port. If you configure flexible Ethernet services encapsulation on the physical interface, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

**flexible-frame-relay**—For IQ interfaces only, use flexible Frame Relay encapsulation when you want to configure multiple per-unit Frame Relay encapsulations. This encapsulation type allows you to configure any combination of TCC, CCC, and standard Frame Relay encapsulations on a single physical port. Also, each logical interface can have any DLCI value from 1 through 1022.

**frame-relay**—Use Frame Relay encapsulation.

**frame-relay-ccc**—Use Frame Relay encapsulation on CCC circuits.

**frame-relay-ether-type**—Use Frame Relay ether type encapsulation for compatibility with the Cisco Frame Relay.

**frame-relay-ether-type-tcc**—Use Frame Relay ether type TCC for Cisco-compatible Frame Relay on TCC circuits to connect different media.

**frame-relay-port-ccc**—Use Frame Relay port CCC encapsulation to transparently carry all the DLCIs between two customer edge (CE) routers without explicitly configuring each DLCI on the two provider edge (PE) routers with Frame Relay transport. When you use this encapsulation type, you can configure the **ccc** family only.

**frame-relay-tcc**—Use Frame Relay encapsulation on TCC circuits to connect different media.

**generic-services**—Use generic services encapsulation for services with a hierarchical scheduler.

**multilink-frame-relay-uni-nni**—Use MLFR UNI NNI encapsulation. This encapsulation is used on link services, voice services interfaces functioning as FRF.16 bundles, and their constituent T1 or E1 interfaces, and is supported on LSQ and redundant LSQ interfaces.

**ppp**—Use serial PPP encapsulation.

**ppp-ccc**—Use serial PPP encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only.

**ppp-tcc**—Use serial PPP encapsulation on TCC circuits for connecting different media. When you use this encapsulation type, you can configure the **tcc** family only.

**vlan-ccc**—Use Ethernet VLAN encapsulation on CCC circuits.

**vlan-vci-ccc**—Use ATM-to-Ethernet interworking encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only. All logical interfaces configured on the Ethernet interface must also have the encapsulation type set to **vlan-vci-ccc**.

**vlan-vpls**—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.



#### NOTE:

- Label-switched interfaces (LSIs) do not support VLAN VPLS encapsulation. Therefore, you can only use VLAN VPLS encapsulation on a PE-router-to-CE-router interface and not a core-facing interface.
- Starting with Junos OS release 13.3, a commit error occurs when you configure **vlan-vpls** encapsulation on a physical interface and configure **family inet** on one of the logical units. Previously, it was possible to commit this invalid configuration.

<b>Required Privilege Level</b>	interface—To view this statement in the configuration.
	interface-control—To add this statement to the configuration.

**Related  
Documentation**

- *Configuring Interface Encapsulation on Physical Interfaces*
- *Configuring CCC Encapsulation for Layer 2 VPNs*
- *Configuring Layer 2 Switching Cross-Connects Using CCC*
- *Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits*
- *Configuring ATM Interface Encapsulation*
- *Configuring ATM-to-Ethernet Interworking*
- *Configuring VLAN Encapsulation*
- *Configuring Extended VLAN Encapsulation*
- *Configuring Encapsulation for Layer 2 Wholesale VLAN Interfaces*
- *Configuring Interfaces for Layer 2 Circuits*
- *Configuring Interface Encapsulation on PTX Series Packet Transport Routers*
- *Configuring MPLS LSP Tunnel Cross-Connects Using CCC*
- *Configuring TCC*
- *Configuring VPLS Interface Encapsulation*
- *Configuring Interfaces for VPLS Routing*
- *Defining the Encapsulation for Switching Cross-Connects*



## evpn

```
Syntax  evpn {
    mac-table-aging-time time;
    designated-forwarder-election-hold-time (evpn)seconds {
        encapsulation-type
        extended-vni-list
        extended-vni-all
        no-default-gateway-ext-comm
    }
    extended-vni-list
    vni-routing-options{
        vni xxx vrf-target export target:xxx:xx
        vni xxx vrf-export name
    }
    vni-routing-options{
        vni xxx vrf-target export target:xxx:xx
        vni xxx vrf-export name
    }
    extended-vlan-list vlan-id | [vlan-id set];
    extended-isid-list (single-isid | isid-list | isid-range | all)
    pbb-evpn-core
    control-word (EVPN)
    interface interface-name{
        ignore-encapsulation-mismatch;
        interface-mac-limit limit {
            packet-action drop;
        }
        no-mac-learning;
        static-mac mac-address;
    }
    interface-mac-limit limit {
        packet-action drop;
    }
    label-allocation per-instance;
    mac-statistics;
    mac-table-size limit {
        packet-action drop;
    }
    no-mac-learning;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier>;
    }
}
```

Hierarchy Level [edit routing-instances *routing-instance-name* protocols]

**Release Information** Statement introduced in Junos OS Release 13.2 for EVPNs on MX 3D Series routers. **designated-forwarder-election-hold-time *seconds*** statement introduced in Junos OS Release 14.1. **extended-vlan-list *vlan-id* | [*vlan-id set*]** statement introduced in Junos OS Release 14.1. Statement introduced in Junos OS Release 14.1-X53-D30 for EVPNs on QFX5100.

<b>Description</b>	Enables an Ethernet VPN (EVPN) on the routing instance.
<b>Options</b>	<b>designated-forwarder-election-hold-time <i>seconds</i></b> —Time in seconds to wait before electing a designated forwarder (DF). <b>Range:</b> 1 through 1800 seconds  The remaining statements are explained separately.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>Configuring EVPN Routing Instances</i></li><li>• <a href="#">Configuring EVPN Routing Instances on EX9200 Switches on page 13</a></li><li>• <a href="#">Tracing EVPN Traffic and Operations on page 15</a></li></ul>

---

## extended-vlan-list

---

<b>Syntax</b>	<code>extended-vlan-list <i>vlan-id</i>   [<i>vlan-id set</i>];</code>
<b>Hierarchy Level</b>	[edit routing-instances <i>routing-instance-name</i> instance-type virtual-switch protocols evpn]
<b>Release Information</b>	Statement introduced in Junos OS Release 14.1. Statement introduced in Junos OS Release 14.2 on EX Series switches.
<b>Description</b>	Specify the VLAN or range of VLANs that are extended over the WAN, wherein all the single VLAN bridge domains corresponding to these VLANs are stretched.
<b>Options</b>	<b><i>vlan-id</i></b> —VLAN ID to be EVPN extended.  <b><i>vlan-id set</i></b> —List of VLAN IDs to be EVPN extended. <b>Range:</b> 1 through 4094 VLANs
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">evpn on page 43</a></li></ul>

## instance-type

<b>Syntax</b>	<code>instance-type type;</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> ], [edit routing-instances <i>routing-instance-name</i> ]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. <b>virtual-switch</b> and <b>layer2-control</b> options introduced in Junos OS Release 8.4. Statement introduced in Junos OS Release 9.2 for EX Series switches. Statement introduced in Junos OS Release 11.3 for the QFX Series. Statement introduced in Junos OS Release 12.3 for ACX Series routers. <b>evpn</b> option introduced in Junos OS Release 13.2 for MX 3D Series routers. <b>forwarding</b> option introduced in Junos OS Release 14.2 for PTX Series Packet Transport Routers.
<b>Description</b>	Define the type of routing instance.

### Options



**NOTE:** On ACX Series routers, you can configure only the forwarding, virtual router, and VRF routing instances.

**type**—Can be one of the following:

- **evpn**—(MX Series routers only) Enable an Ethernet VPN (EVPN) on the routing instance. You cannot configure the **evpn** option under the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* instance-type] hierarchy level.
- **forwarding**—Provide support for filter-based forwarding, where interfaces are not associated with instances. All interfaces belong to the default instance. Other instances are used for populating RPD learned routes. For this instance type, there is no one-to-one mapping between an interface and a routing instance. All interfaces belong to the default instance inet.0.
- **l2backhaul-vpn**—Provide support for Layer 2 wholesale VLAN packets with no existing corresponding logical interface. When using this instance, the router learns both the outer tag and inner tag of the incoming packets, when the **instance-role** statement is defined as **access**, or the outer VLAN tag only, when the **instance-role** statement is defined as **nni**.
- **l2vpn**—Enable a Layer 2 VPN on the routing instance. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.
- **layer2-control**—(MX Series routers only) Provide support for RSTP or MSTP in customer edge interfaces of a VPLS routing instance. This instance type cannot be used if the customer edge interface is multihomed to two provider edge interfaces. If the customer

edge interface is multihomed to two provider edge interfaces, use the default BPDU tunneling.

- **no-forwarding**—This is the default routing instance. Do not create a corresponding forwarding instance. Use this routing instance type when a separation of routing table information is required. There is no corresponding forwarding table. All routes are installed into the default forwarding table. IS-IS instances are strictly nonforwarding instance types.
- **virtual-router**—Enable a virtual router routing instance. This instance type is similar to a VPN routing and forwarding instance type, but used for non-VPN-related applications. You must configure the **interface** statement for this type of routing instance. You do not need to configure the **route-distinguisher**, **vrf-import**, and **vrf-export** statements.
- **virtual-switch**—(MX Series routers and EX9200 switches only) Provide support for Layer 2 bridging. Use this routing instance type to isolate a LAN segment with its Spanning Tree Protocol (STP) instance and to separate its VLAN identifier space.
- **vpls**—Enable VPLS on the routing instance. Use this routing instance type for point-to-multipoint LAN implementations between a set of sites in a VPN. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.
- **vrf**—VPN routing and forwarding (VRF) instance. Provides support for Layer 3 VPNs, where interface routes for each instance go into the corresponding forwarding table only. Required to create a Layer 3 VPN. Create a VRF table (*instance-name.inet.0*) that contains the routes originating from and destined for a particular Layer 3 VPN. For this instance type, there is a one-to-one mapping between an interface and a routing instance. Each VRF instance corresponds with a forwarding table. Routes on an interface go into the corresponding forwarding table. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.

<b>Required Privilege</b>	routing—To view this statement in the configuration.
<b>Level</b>	routing-control—To add this statement to the configuration.

- |                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Related Documentation</b> | <ul style="list-style-type: none"><li>• <i>Configuring Routing Instances on PE Routers in VPNs</i></li><li>• <i>Configuring EVPN Routing Instances</i></li><li>• <a href="#">Configuring EVPN Routing Instances on EX9200 Switches on page 13</a></li><li>• <i>Configuring Virtual Router Routing Instances</i></li><li>• <i>Example: Configuring Filter-Based Forwarding on the Source Address</i></li><li>• <i>Example: Configuring Filter-Based Forwarding on Logical Systems</i></li><li>• <i>Layer 2 Routing Instance Types</i></li></ul> |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## interface (EVPN Routing Instances)

<b>Syntax</b>	<pre> interface <i>interface-name</i> {   ignore-encapsulation-mismatch;   interface-mac-limit <i>limit</i> {     packet-action drop;   }   no-mac-learning;   static-mac <i>mac-address</i>; } </pre>
<b>Hierarchy Level</b>	[edit routing-instances <i>routing-instance-name</i> protocols evpn]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2 for EVPNs on MX 3D Series routers. Statement introduced in Junos OS Release 14.2 on EX Series switches.
<b>Description</b>	Specify each interface over which the Ethernet VPN (EVPN) traffic travels between the PE device and customer edge (CE) device. The interfaces are bound to the EVPN routing instance.
<b>Options</b>	<p><i>interface-name</i>—Name of the interface.</p> <p>The remaining statements are explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring EVPN Routing Instances</a></li> <li>• <a href="#">Configuring EVPN Routing Instances on EX9200 Switches on page 13</a></li> <li>• <a href="#">evpn on page 43</a></li> <li>• <a href="#">instance-type on page 45</a></li> </ul>

## interface (Routing Instances)

---

<b>Syntax</b>	<code>interface <i>interface-name</i> {     description <i>text</i>; }</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> ], [edit routing-instances <i>routing-instance-name</i> ]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.2 for EX Series switches. Statement introduced in Junos OS Release 12.3 for ACX Series routers. Statement introduced in Junos OS Release 13.2 for MX 3D Series routers.
<b>Description</b>	Specify the interface over which the VPN traffic travels between the PE device and CE device. You configure the interface on the PE device. If the value <b>vrf</b> is specified for the <b>instance-type</b> statement included in the routing instance configuration, this statement is required.
<b>Options</b>	<i>interface-name</i> —Name of the interface.  The remaining statement is explained separately.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Configuring Routing Instances on PE Routers in VPNs</a></li><li>• <a href="#">Configuring EVPN Routing Instances</a></li><li>• <a href="#">Configuring EVPN Routing Instances on EX9200 Switches on page 13</a></li><li>• <a href="#">interface (VPLS Routing Instances)</a></li></ul>

## interface-mac-limit (VPLS)

<b>Syntax</b>	<pre>interface-mac-limit <i>limit</i> {     packet-action drop; }</pre>
<b>Hierarchy Level</b>	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls site <i>site-name</i> interfaces <i>interface-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols evpn],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols evpn interface <i>interface-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls site <i>site-name</i> interfaces <i>interface-name</i>]</p>
<b>Release Information</b>	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Support for EVPNs introduced in Junos OS Release 13.2 on MX 3D Series routers.</p> <p>Support for EVPNs introduced in Junos OS Release 14.2 on EX Series switches.</p>
<b>Description</b>	<p>Specify the maximum number of media access control (MAC) addresses that can be learned by the EVPN or VPLS routing instance. You can configure the same limit for all interfaces configured for a routing instance. You can also configure a limit for a specific interface.</p> <p>Starting with Junos OS Release 12.3R4, if you do not configure the parameter to limit the number of MAC addresses to be learned by a VPLS instance, the default value is not effective. Instead, if you do not include the <b>interface-mac-limit</b> option at the <b>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls site <i>site-name</i> interfaces <i>interface-name</i>]</b>, hierarchy level, this setting is not present in the configuration with the default value of 1024 addresses. If you upgrade a router running a Junos OS release earlier than Release 12.3R4 to Release 12.3R4 or later, you must configure the <b>interface-mac-limit</b> option with a valid value for it to be saved in the configuration.</p>
<b>Options</b>	<p><b>limit</b>—Number of MAC addresses that can be learned from each interface.</p> <p><b>Range:</b> 16 through 65,536 MAC addresses</p> <p><b>Default:</b> 1024 addresses</p> <p>The remaining statement is explained separately.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring EVPN Routing Instances</a></li> <li>• <a href="#">Configuring EVPN Routing Instances on EX9200 Switches on page 13</a></li> <li>• <a href="#">Configuring VPLS Routing Instances</a></li> <li>• <a href="#">interface on page 47</a></li> <li>• <a href="#">mac-table-size on page 51</a></li> </ul>


## mac-statistics

<b>Syntax</b>	mac-statistics;
<b>Hierarchy Level</b>	<p>[edit bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit logical-systems <i>logical-system-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> switch-options],</p> <p>[edit logical-systems <i>logical-system-name</i> switch-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> switch-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols <a href="#">evpn</a>],</p> <p>[edit switch-options],</p> <p>[edit switch-options],</p> <p>[edit vlans <i>vlan-name</i> switch-options]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.4.</p> <p>Support for the <b>switch-options</b> statement added in Junos OS Release 9.2.</p> <p>Support for top-level configuration for the <b>virtual-switch</b> type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.</p> <p>Support for logical systems added in Junos OS Release 9.6.</p> <p>[edit switch-options] and [edit vlans <i>vlan-name</i> switch-options] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.</p> <p>Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.</p> <p>[edit switch-options] and [edit vlans <i>vlan-name</i> switch-options] hierarchy levels introduced in Junos OS Release 13.2 for the QFX Series.</p>
<b>Description</b>	(MX Series routers, EX Series switches, and QFX Series only) For bridge domains or VLANs, enable MAC accounting either for a specific bridge domain or VLAN, or for a set of bridge domains or VLANs associated with a Layer 2 trunk port.
<b>Default</b>	disabled
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Layer 2 Learning and Forwarding for Bridge Domains Overview</i></li> <li>• <i>Layer 2 Learning and Forwarding for VLANs Overview</i></li> <li>• <i>Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports</i></li> <li>• <i>Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port</i></li> <li>• <i>Configuring EVPN Routing Instances</i></li> </ul>



- [Configuring EVPN Routing Instances on EX9200 Switches on page 13](#)

## mac-table-size

<b>Syntax</b>	<code>mac-table-size size {     <code>packet-action</code> drop; }</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls], [edit routing-instances <i>routing-instance-name</i> protocols evpn], [edit routing-instances <i>routing-instance-name</i> protocols vpls]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 13.2 for EVPNs on MX 3D Series routers. Statement introduced in Junos OS Release 14.2 for EX Series switches.
<b>Description</b>	Specify the size of the MAC address table.
<b>Options</b>	<p><i>size</i>—Size of the MAC address table.</p> <p><b>Range:</b></p> <ul style="list-style-type: none"> <li>• (M Series and T Series routers only) 16 through 65,536 MAC addresses</li> <li>• (MX Series routers only) 16 through 1,048,575 MAC addresses</li> <li>• (T4000 routers with Type 5 FPCs only) 16 through 262,143 MAC addresses</li> </ul>
<div>  <p><b>NOTE:</b> Before modifying the size of the MAC address table (to 262,143 addresses), you must enable network services mode by including the <code>enhanced-mode</code> statement at the [edit chassis network-services] hierarchy level and then reboot the router.</p> </div>	
<b>Default:</b>	512 MAC addresses
	The remaining statement is explained separately.
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuring EVPN Routing Instances</a></li> <li>• <a href="#">Configuring EVPN Routing Instances on EX9200 Switches on page 13</a></li> <li>• <a href="#">Configuring VPLS Routing Instances</a></li> <li>• <a href="#">Configuring Improved VPLS MAC Address Learning on T4000 Routers with Type 5 FPCs</a></li> <li>• <a href="#">enhanced-mode</a></li> <li>• <a href="#">evpn on page 43</a></li> </ul>

## no-mac-learning

<b>Syntax</b>	no-mac-learning;
<b>Hierarchy Level</b>	<p>[edit bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit bridge-domains <i>bridge-domain-name</i> bridge-options interface <i>interface-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit logical-systems <i>logical-system-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options interface <i>interface-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options interface <i>interface-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> switch-options],</p> <p>[edit logical-systems <i>logical-system-name</i> switch-options],</p> <p>[edit bridge-domains <i>bridge-domain-name</i> bridge-options interface <i>interface-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options interface <i>interface-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols <a href="#">evpn</a>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols evpn <a href="#">interface</a> <i>interface-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> switch-options],</p> <p>[edit switch-options],</p> <p>[edit switch-options interface <i>interface-name</i>],</p> <p>[set vlans <i>vlan-name</i> switch-options]</p>
<b>Release Information</b>	<p>Statement introduced in Junos OS Release 8.4.</p> <p>Support for the <b>switch-options</b> statement added in Junos OS Release 9.2.</p> <p>Support for top-level configuration for the <b>virtual-switch</b> type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or bridge domain configured within a virtual switch.</p> <p>Support for logical systems added in Junos OS Release 9.6.</p> <p>[<a href="#">edit switch-options</a>], [<a href="#">edit switch-options interface <i>interface-name</i></a>], [<a href="#">edit vlans <i>vlan-name</i> switch-options</a>], and [<a href="#">edit vlans <i>vlan-name</i> switch-options interface <i>interface-name</i></a>] hierarchy levels introduced in Junos OS Release 12.3 R2 for EX Series switches.</p> <p>Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.</p> <p>Hierarchy levels [<a href="#">edit switch-options interface <i>interface-name</i></a>] and [<a href="#">edit vlans <i>vlan-name</i> switch-options</a>] introduced in Junos OS Release 13.2X50-D10 for EX Series switches.</p>
<b>Description</b>	<p>For MX Series routers and EX Series switches, disable MAC learning for a virtual switch, for a bridge domain or VLAN, for a specific logical interface in a bridge domain or VLAN, or for a set of bridge domains or VLANs associated with a Layer 2 trunk port. On platforms that support EVPNs, you can disable MAC learning on an EVPN.</p>



**NOTE:** When MAC learning is disabled for a VPLS routing instance, traffic is not load-balanced and only one of the equal-cost next hops is used.

<b>Default</b>	MAC learning is enabled.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Configuring EVPN Routing Instances</i></li> <li>• <a href="#">Configuring EVPN Routing Instances on EX9200 Switches on page 13</a></li> <li>• <i>Layer 2 Learning and Forwarding for Bridge Domains Overview</i></li> <li>• <i>Layer 2 Learning and Forwarding for VLANs Overview</i></li> <li>• <i>Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports</i></li> <li>• <i>Understanding Bridging and VLANs on EX Series Switches</i></li> <li>• <i>Understanding Q-in-Q Tunneling on EX Series Switches</i></li> </ul>

## packet-action

**Syntax** `packet-action action;`

**Hierarchy Level** [edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],  
 [edit bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],  
 [edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],  
 [edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],  
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],  
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],  
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface *interface-name* interface-mac-limit *limit*],  
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface-mac-limit *limit*],  
 [edit logical-systems *logical-system-name* switch-options interface-mac-limit *limit*],  
 [edit protocols l2-learning global-mac-limit *limit*],  
 [edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],  
 [edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],  
 [edit routing-instances *routing-instance-name* protocols evpn **interface-mac-limit (VPLS)**],  
 [edit routing-instances *routing-instance-name* protocols evpn interface *interface-name* **interface-mac-limit (VPLS)**],  
 [edit routing-instances *routing-instance-name* protocols evpn **mac-table-size limit**],  
 [edit routing-instances *routing-instance-name* switch-options interface *interface-name* interface-mac-limit *limit*],  
 [edit routing-instances *routing-instance-name* switch-options interface-mac-limit *limit*],  
 [edit switch-options interface-mac-limit *limit*],  
 [edit switch-options interface *interface-name* interface-mac-limit *limit*],  
 [edit switch-options interface-mac-limit *limit*],  
 [edit switch-options interface *interface-name* interface-mac-limit *limit*],  
 [edit switch-options interface-mac-limit *limit*],  
 [edit switch-options mac-table-size *limit*],  
 [edit switch-options interface *interface-name* interface-mac-limit *limit*],  
 [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*],  
 [edit vlans *vlan-name* switch-options interface-mac-limit *limit*],  
 [edit vlans *vlan-name* switch-options mac-table-size *limit*],  
 [edit vlans *vlan-name* switch-options interface-mac-limit *limit*],  
 [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*],  
 [edit vlans *vlan-name* switch-options mac-table-size *limit*]

**Release Information** Statement introduced in Junos OS Release 8.4.  
 Support for the **switch-options** statement added in Junos OS Release 9.2.  
 Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.

Support for logical systems added in Junos OS Release 9.6.

[edit switch-options interface *interface-name* interface-mac-limit *limit*], [edit switch-options interface-mac-limit *limit*], [edit switch-options mac-table-size *limit*], [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*], [edit vlans *vlan-name* switch-options interface-mac-limit *limit*], and [edit vlans *vlan-name* switch-options mac-table-size *limit*] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.

Support for EVPNs introduced in Junos OS Release 13.2 on MX Series 3D Universal Edge Routers.

Support at the [edit switch-options interface *interface-name* interface-mac-limit *limit*] hierarchy level and hierarchy levels under [edit vlans *vlan-name*] introduced in Junos OS Release 13.2X50-D10 for EX Series switches and Junos OS Release 13.2 for the QFX Series.

**Description** Specify the action taken when packets with new source MAC addresses are received after the MAC address limit is reached. If this statement is not configured, packets with new source MAC addresses are forwarded by default.

**Default**



**NOTE:** On a QFX Series Virtual Chassis, if you include the shutdown option at the [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit packet-action] hierarchy level and issue the commit operation, the system generates a commit error. The system does not generate an error if you include the shutdown option at the [edit switch-options interface *interface-name* interface-mac-limit packet-action] hierarchy level.

Disabled. The default is for packets for new source MAC addresses to be forwarded after the MAC address limit is reached.

**Options**

**drop**—Drop packets with new source MAC addresses, and do not learn the new source MAC addresses.

**drop-and-log**—(EX Series switches and QFX Series only) Drop packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

**log**—(EX Series switches and QFX Series only) Hold packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

**none**—(EX Series switches and QFX Series only) Forward packets with new source MAC addresses, and learn the new source MAC address.

**shutdown**—(EX Series switches and QFX Series only) Disable the specified interface, and generate an alarm, an SNMP trap, or a system log entry.

**Required Privilege**

routing—To view this statement in the configuration.

**Level**

routing-control—To add this statement to the configuration.

- Related Documentation**
- [Configuring EVPN Routing Instances](#)
  - [Configuring EVPN Routing Instances on EX9200 Switches on page 13](#)
  - [Configuring MAC Limiting \(CLI Procedure\)](#)
  - [Configuring Persistent MAC Learning \(CLI Procedure\)](#)
  - [Layer 2 Learning and Forwarding for Bridge Domains Overview](#)
  - [Layer 2 Learning and Forwarding for VLANs Overview](#)
  - [Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports](#)
  - [Layer 2 Learning and Forwarding for VLANs Overview](#)
  - [Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port](#)

---

## routing-instances

---

- Syntax** `routing-instances routing-instance-name { ... }`
- Hierarchy Level** [edit],  
[edit logical-systems *logical-system-name*]
- Release Information** Statement introduced before Junos OS Release 7.4.
- Description** Configure an additional routing entity for a router or switch. You can create multiple instances of BGP, IS-IS, OSPF, OSPF version 3 (OSPFv3), and RIP for a router or switch.
- Default** Routing instances are disabled for the router or switch.
- Options** *routing-instance-name*—Name of the routing instance, a maximum of 31 characters. The remaining statements are explained separately.
- Required Privilege Level** routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.
- Related Documentation**
- [Configuring EVPN Routing Instances](#)
  - [Configuring Routing Instances on PE Routers in VPNs](#)

## traceoptions (Protocols EVPN)

<b>Syntax</b>	<pre>traceoptions {     file <i>filename</i> &lt;files <i>number</i>&gt; &lt;size <i>size</i>&gt; &lt;world-readable   no-world-readable&gt;;     flag <i>flag</i> &lt;flag-modifier&gt;; }</pre>
<b>Hierarchy Level</b>	[edit routing-instances <i>routing-instance-name</i> protocols evpn]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2 for MX 3D Series routers. Statement introduced in Junos OS Release 14.2 for EX Series switches.
<b>Description</b>	Trace traffic flowing through an EVPN routing instance.
<b>Options</b>	<p><b>file <i>filename</i></b>—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").</p> <p><b>files <i>number</i></b>—(Optional) Maximum number of trace files. When a trace file named <b><i>trace-file</i></b> reaches the maximum size as specified by the <b>size</b> option, it is renamed <b><i>trace-file.0</i></b>. When <b><i>trace-file</i></b> again reaches the maximum size, <b><i>trace-file.0</i></b> is renamed <b><i>trace-file.1</i></b> and <b><i>trace-file</i></b> is renamed <b><i>trace-file.0</i></b>. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.</p> <p>If you specify a maximum number of files, you also must specify a maximum file size with the <b>size</b> option.</p> <p><b>Range:</b> 2 through 1000 files</p> <p><b>Default:</b> 2 files</p> <p><b>flag <i>flag</i></b>—Tracing operation to perform. To specify more than one tracing operation, include multiple <b>flag</b> statements. You can specify the following tracing flags:</p> <ul style="list-style-type: none"> <li>• <b>all</b>—All EVPN tracing options</li> <li>• <b>error</b>—Error conditions</li> <li>• <b>general</b>—General events</li> <li>• <b>mac-database</b>—MAC route database in the EVPN routing instance</li> <li>• <b>nlri</b>—EVPN advertisements received or sent by means of BGP</li> <li>• <b>normal</b>—Normal events</li> <li>• <b>oam</b>—OAM messages</li> <li>• <b>policy</b>—Policy processing</li> <li>• <b>route</b>—Routing information</li> <li>• <b>state</b>—State transitions</li> <li>• <b>task</b>—Routing protocol task processing</li> <li>• <b>timer</b>—Routing protocol timer processing</li> </ul>

- **topology**—EVPN topology changes caused by reconfiguration or advertisements received from other provider edge (PE) routers using BGP

**flag-modifier**—(Optional) Modifier for the tracing flag. You can specify the following modifiers:

- **detail**—Provide detailed trace information.
- **disable**—Disable this trace flag.
- **receive**—Trace received packets.
- **send**—Trace sent packets.

**no-world-readable**—Do not allow any user to read the log file.

**size size**—(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 **trace-file** again reaches the 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 (as specified by the **files** option) is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

**Syntax:** **xk** to specify kilobytes, **xm** to specify megabytes, or **xg** to specify gigabytes

**Range:** 10 KB through the maximum file size supported on your system

**Default:** 1 MB

**world-readable**—Allow any user to read the log file.

<b>Required Privilege Level</b>	routing—To view this statement in the configuration.
	routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Tracing EVPN Traffic and Operations on page 15</a></li></ul>



## vlan-id (routing instance)

<b>Syntax</b>	<code>vlan-id (vlan-id   all   none);</code>
<b>Hierarchy Level</b>	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> ], [edit routing-instances <i>routing-instance-name</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 13.2. Statement introduced in Junos OS Release 14.2 for EX Series switches.
<b>Description</b>	Specify 802.1Q VLAN tag IDs to a routing instance.
<b>Options</b>	<p><b>vlan-id</b>—A valid VLAN identifier.</p> <p><b>Range:</b> For 4-port Fast Ethernet PICs, 512 through 1023. For 1-port and 10-port Gigabit Ethernet PICs configured to handle VPLS traffic, 512 through 4094.</p> <p><b>all</b>—Include all VLAN identifiers specified on the logical interfaces included in the routing instance.</p> <p><b>none</b>—Include no VLAN identifiers for the routing instance.</p>
<b>Required Privilege Level</b>	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Configuring EVPN Routing Instances</i></li> <li>• <a href="#">Configuring EVPN Routing Instances on EX9200 Switches on page 13</a></li> </ul>



## CHAPTER 7

# Operational Commands

- `show evpn database`
- `show evpn flood`
- `show evpn instance`
- `show evpn mac-table`
- `show evpn peer-gateway-macs`
- `show evpn statistics`

## show evpn database

---

<b>Syntax</b>	<code>show evpn database</code> <code>&lt; extensive &gt;</code> <code>&lt; instance <i>instance-name</i> &gt;</code> <code>&lt; interface <i>interface-name</i> &gt;</code> <code>&lt; mac-address <i>address</i> &gt;</code> <code>&lt; neighbor <i>neighbor-name</i> &gt;</code> <code>&lt; origin <i>origin-name</i> &gt;</code> <code>&lt; state <i>state-name</i> &gt;</code> <code>&lt; vlan-id <i>vlan-id</i> &gt;</code>
<b>Release Information</b>	Command introduced in Junos OS Release 14.2 for EX Series switches.
<b>Description</b>	Show Ethernet VPN (EVPN) database information.
<b>Options</b>	<p><b>none</b>—Display brief information about the EVPN database.</p> <p><b>extensive</b> —(Optional) Display the specified level of output.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Display MAC addresses from a specific routing instance.</p> <p><b>interface <i>interface-name</i></b>—(Optional) Display the MAC address learned from the specified interface.</p> <p><b>mac-address <i>address</i></b>—(Optional) Display the specified MAC address.</p> <p><b>neighbor <i>neighbor-name</i></b>—(Optional) Display the MAC address learned from the specified neighbor.</p> <p><b>origin <i>origin-name</i></b>—(Optional) Display the MAC address with the specified origin.</p> <p><b>state <i>state-name</i></b>—(Optional) Display the MAC address with the specified state.</p> <p><b>vlan-id <i>vlan-id</i></b>—(Optional) Display the MAC address with the specified VLAN.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 23</a></li></ul>

## show evpn flood

---

<b>Syntax</b>	<pre>show evpn flood event-queue &lt;instance <i>instance-name</i>&gt; &lt;logical-system &lt;<i>logical-system-name</i>&gt;&gt; &lt;route <i>route-name</i>&gt;</pre>
<b>Release Information</b>	Command introduced in Junos OS Release 14.2 for EX Series switches.
<b>Description</b>	Show Ethernet VPN (EVPN) flooding information.
<b>Options</b>	<p><b>none</b>—Display brief information about EVPN flooding.</p> <p><b>brief   detail   extensive   summary</b>—(Optional) Display the specified level of output.</p> <p><b>event-queue</b>—(Optional) Display the queue of pending EVPN flood events.</p> <p><b>instance <i>instance-name</i></b>—(Optional) Display flooding information for the specified routing instance.</p> <p><b>logical-system &lt;<i>logical-system-name</i>&gt;</b>—(Optional) Display flooding information for the specified logical system or all logical systems.</p> <p><b>route <i>route-name</i></b>—(Optional) Display flooding information for the specified route.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 23</a></li> </ul>

## show evpn instance

<b>Syntax</b>	<pre>show evpn instance &lt;brief   extensive&gt; &lt;backup-forwarder&gt; &lt;designated-forwarder&gt; &lt;esi esi&gt; &lt;instance-name&gt; &lt;neighbor neighbor-address&gt;</pre>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 14.1.</p> <p>Command introduced in Junos OS Release 14.2 for EX Series switches.</p>
<b>Description</b>	Show Ethernet VPN (EVPN) routing instance information.
<b>Options</b>	<p><b>none</b>—Display brief information about the EVPN routing instance.</p> <p><b>brief   extensive</b>—(Optional) Display the specified level of output.</p> <p><b>backup-forwarder</b>—(Optional) Display IP addresses of all the backup designated forwarder (non-DFs) for the Ethernet segment.</p> <p><b>designated-forwarder</b>—(Optional) Display IP address of the designated forwarder for the Ethernet segment.</p> <p><b>esi esi</b>—(Optional) Display brief information about the routing instance associated with this ESI value.</p> <p><b>instance-name</b>—(Optional) Display information for the specified routing instance.</p> <p><b>neighbor neighbor-address</b>—(Optional) Display IP address of the EVPN neighbor.</p>
<b>Required Privilege Level</b>	view
<b>List of Sample Output</b>	<p><a href="#">show evpn instance brief on page 66</a></p> <p><a href="#">show evpn instance on page 66</a></p> <p><a href="#">show evpn instance extensive on page 67</a></p> <p><a href="#">show evpn instance instance-name esi esi-value backup-forwarder on page 68</a></p> <p><a href="#">show evpn instance instance-name esi esi-value designated-forwarder on page 68</a></p>
<b>Output Fields</b>	<p><a href="#">Table 3 on page 64</a> lists the output fields for the <b>show evpn instance</b> command. Output fields are listed in the approximate order in which they appear.</p>

**Table 3: show evpn instance Output Fields**

Field Name	Field Description	Level of Output
<b>Instance</b>	Names of the routing instances.	All Levels
<b>Intfs</b>	Total number of interfaces participating in each routing instance, and number of interfaces that are up.	<b>brief</b>

Table 3: show evpn instance Output Fields (*continued*)

Field Name	Field Description	Level of Output
IRB intfs	Statistics on number of IRB interfaces for each routing instance: <ul style="list-style-type: none"> <li>• <b>Total</b>—Total number of IRB interfaces.</li> <li>• <b>Up</b>—Number of active IRB interfaces.</li> <li>• <b>Nbrs</b>—Number of neighbor IRB interfaces.</li> </ul>	<b>brief</b>
MH ESIs	Number of Ethernet segments per routing instance that connect to a multihomed customer site.	<b>brief</b>
MAC addresses	Number of local and remote MAC addresses for each routing instance.	<b>brief</b>
Route Distinguisher	Unique route distinguisher associated with this routing instance.	<b>extensive</b>
Per-instance MAC route label	Label of MAC route for each routing instance.	<b>extensive</b>
Per-instance multicast route label	Label of multicast route for each routing instance.	<b>extensive</b>
Total MAC addresses	Total number of MAC addresses received for each routing instance.	<b>extensive</b>
Default gateway MAC addresses	Number of MAC addresses serving as a default gateway in the routing instance.	<b>extensive</b>
Number of local interfaces	Number of local interfaces belonging to this routing instance.	<b>extensive</b>
Interface name	Name of interfaces that belong to this routing instance.	<b>extensive</b>
ESI	Ethernet segment identifier (ESI) value of the interfaces belonging to this routing instance.	<b>extensive</b>
Mode	Mode of operation for each routing instance: <ul style="list-style-type: none"> <li>• <b>single-homed</b>—Default mode and does not require Ethernet segment values to be configured.</li> <li>• <b>single-active</b>—EVPN active-standby multihoming mode of operation.</li> </ul>	<b>extensive</b>
SH label	Split horizon label used for the active-standby multihoming mode of operation.	<b>extensive</b>
Number of IRB interfaces	Number of IRB interfaces that belong to this routing instance.	<b>extensive</b>

Table 3: show evpn instance Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>L3 context</b>	Names of routing instances that have the Layer 3 routes installed for an EVPN IRB interface, typically a VRF routing instance.	<b>extensive</b>
<b>Number of neighbors</b>	Number of neighbors connected to this routing instance and their IP addresses.	<b>extensive</b>
<b>MAC address advertisement</b>	Number of MAC address advertisements received from the neighbor.	<b>extensive</b>
<b>MAC+IP address advertisement</b>	Number of MAC and IP address advertisements received from the neighbor.	<b>extensive</b>
<b>Inclusive multicast</b>	Number of inclusive multicast routes received from the neighbor.	<b>extensive</b>
<b>Ethernet auto-discovery</b>	Number of autodiscovery routes per Ethernet segment received from the neighbor.	<b>extensive</b>
<b>Number of ethernet segments</b>	Total number of Ethernet segments for the routing instance.	<b>extensive</b>
<b>Designated forwarder</b>	IP address of the designated forwarder (DF) for the Ethernet segment.	<b>extensive</b>
<b>Backup forwarder</b>	IP address of all the backup designated forwarders (BDFs) or non-DF routers for the Ethernet segment.  <b>NOTE:</b> Immediately after an evpn interface esi value is changed and the new configuration is committed, the <b>Designated forwarder</b> information will change to <b>DF not elected yet</b> and the Backup forwarder information will not be displayed until after the election is complete.	<b>extensive</b>

## Sample Output

### show evpn instance brief

```

user@host> show evpn instance brief

```

Instance	Intfs		IRB intfs		MH		MAC addresses		
	Total	Up	Total	Up	Nbrs	ESIs	Local	Remote	
ALPHA	2	2	1	1	2	1	3	4	
BETA	2	2	1	1	2	1	2	4	
__default__evpn__	0	0	0	0	1	0	0	0	

### show evpn instance

```

user@host> show evpn instance
Instance: black
Route Distinguisher: 101:101
VLAN ID: 100
Label allocation mode: Per-instance

```



```

Per-instance MAC route label: 299776
Per-instance multicast route label: 299792
Number of local interfaces: 1
Number of local interfaces up: 1
  Interface name    Static MACs    ESI
  cbp-0.0           0          0
Number of neighbors: 1
  55.55.55.1
    Received routes
      MAC address advertisement:      1
      Ethernet auto-discovery:        0
      Inclusive multicast:             1

```

### show evpn instance extensive

```

user@host> show evpn instance extensive
Instance: ALPHA
  Route Distinguisher: 10.255.0.1:100
  Per-instance MAC route label: 300144
  Per-instance multicast route label: 300160
  MAC database status
    Local Remote
  Total MAC addresses:      3      4
  Default gateway MAC addresses: 1      2
  Number of local interfaces: 2 (2 up)
    Interface name  ESI                               Mode          SH label
  ae0.0            00:11:22:33:44:55:66:77:88:99    single-active
  ge-0/0/2.0       00:00:00:00:00:00:00:00:00:00    single-homed
  Number of IRB interfaces: 1 (1 up)
    Interface name  L3 context
  irb.0            DELTA
  Number of neighbors: 2
    10.255.0.2
      Received routes
        MAC address advertisement:      2
        MAC+IP address advertisement:    3
        Inclusive multicast:             1
        Ethernet auto-discovery:         1
    10.255.0.3
      Received routes
        MAC address advertisement:      2
        MAC+IP address advertisement:    2
        Inclusive multicast:             1
        Ethernet auto-discovery:         0
  Number of ethernet segments: 1
    ESI: 00:11:22:33:44:55:66:77:88:99
    Designated forwarder: 10.255.0.1
    Backup forwarder: 10.255.0.2

Instance: BETA
  Route Distinguisher: 10.255.0.1:300
  VLAN ID: 300
  Per-instance MAC route label: 300176
  Per-instance multicast route label: 300192
  MAC database status
    Local Remote
  Total MAC addresses:      3      4
  Default gateway MAC addresses: 1      2
  Number of local interfaces: 2 (2 up)
    Interface name  ESI                               Mode          SH label
  ae1.0            00:00:00:00:00:00:00:00:00:00    single-homed
  ge-0/0/4.0       00:22:44:66:88:00:22:44:66:88    single-active
  Number of IRB interfaces: 1 (1 up)

```

```

Interface name  L3 context
irb.1          DELTA
Number of neighbors: 2
10.255.0.2
  Received routes
    MAC address advertisement:      2
    MAC+IP address advertisement:   3
    Inclusive multicast:            1
    Ethernet auto-discovery:        1
10.255.0.3
  Received routes
    MAC address advertisement:      2
    MAC+IP address advertisement:   2
    Inclusive multicast:            1
    Ethernet auto-discovery:        0
Number of ethernet segments: 1
ESI: 00:22:44:66:88:00:22:44:66:88
Designated forwarder: 10.255.0.1
Backup forwarder: 10.255.0.2

Instance: __default_evpn__
Route Distinguisher: 10.255.0.1:0
VLAN ID: 0
Per-instance MAC route label: 300208
Per-instance multicast route label: 300224
MAC database status          Local Remote
Total MAC addresses:         0       0
Default gateway MAC addresses: 0       0
Number of local interfaces: 0 (0 up)
Number of IRB interfaces: 0 (0 up)
Number of neighbors: 1
10.255.0.2
  Received routes
    Ethernet auto-discovery:        0
    Ethernet Segment:               2
Number of ethernet segments: 0

```

#### show evpn instance instance-name esi esi-value backup-forwarder

```

user@host> show evpn instance ALPHA esi 00:11:22:33:44:55:66:77:88:99 backup-forwarder
Instance: ALPHA
Number of ethernet segments: 1
ESI: 00:11:22:33:44:55:66:77:88:99
Backup forwarder: 10.255.0.2

```

#### show evpn instance instance-name esi esi-value designated-forwarder

```

user@host> show evpn instance ALPHA esi 00:11:22:33:44:55:66:77:88:99 designated-forwarder
Instance: ALPHA
Number of ethernet segments: 1
ESI: 00:11:22:33:44:55:66:77:88:99
Designated forwarder: 10.255.0.1

```

## show evpn mac-table

<b>Syntax</b>	<pre>show evpn mac-table &lt;address&gt; &lt;brief   count   detail   extensive   summary&gt; &lt;instance instance-name&gt; &lt;interface interface-name&gt; &lt;isid &lt;isid&gt;&gt; &lt;logical-system &lt;logical-system-name&gt;&gt; &lt;vlan-id vlan-id&gt;</pre>
<b>Release Information</b>	Command introduced in Junos OS Release 14.2 for EX Series switches.
<b>Description</b>	Show Ethernet VPN (EVPN) MAC table information.
<b>Options</b>	<p><b>none</b>—Display brief information about the EVPN MAC table.</p> <p><b>address</b>—(Optional) Display MAC table information for the specified MAC address.</p> <p><b>brief   count   detail   extensive   summary</b>—(Optional) Display the specified level of output.</p> <p><b>instance instance-name</b>—(Optional) Display MAC table information for a specific routing instance.</p> <p><b>isid &lt;isid&gt;</b>—(Optional) Display MAC table information for the specified ISID or all ISIDs.</p> <p><b>logical-system &lt;logical-system-name&gt;</b>—(Optional) Display MAC table information for the specified logical system or all logical systems.</p> <p><b>vlan-id vlan-id</b>—(Optional) Display MAC table information for the specified VLAN.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 23</a></li> </ul>

## show evpn peer-gateway-macs

---

<b>Syntax</b>	show evpn peer-gateway-macs <address> <instance <i>instance-name</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 14.2 for EX Series switches.
<b>Description</b>	Show Ethernet VPN (EVPN) peer gateway MAC information.
<b>Options</b>	<b>none</b> —Display brief information about the EVPN peer gateway MAC.  <b>address</b> —(Optional) Display peer gateway information for the specified MAC address.  <b>instance <i>instance-name</i></b> —(Optional) Display peer gateway MAC information for the specified routing instance.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 23</a></li></ul>

## show evpn statistics

---

<b>Syntax</b>	show evpn statistics <instance <i>instance-name</i> >
<b>Release Information</b>	Command introduced in Junos OS Release 14.2 for EX Series switches.
<b>Description</b>	Show Ethernet VPN (EVPN) statistics information.
<b>Options</b>	<b>none</b> —Display brief information about EVPN statistics.  <b>instance <i>instance-name</i></b> —(Optional) Display EVPN statistics for the specified routing instance.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 23</a></li></ul>

