HP Switch Software

IPv6 Configuration Guide

2530-48G-PoE+
2530-48-PoE+
2530-24G-PoE+
2530-24-PoE+
2530-8G-PoE+
2530-8-PoE+
2530-48G
2530-48
2530-24G
2530-24
2530-8G
2530-8

Software version YA/YB.15.12
May 2013

HP Series 2530 Switches

Abstract

This switch software guide is intended for network administrators and support personnel, and applies to the switch models listed on this page unless otherwise noted. This guide does not provide information about upgrading or replacing switch hardware. The information in this guide is subject to change without notice.

May 2013
YA/YB.15.12

IPv6 Configuration Guide

Publication Number

5998-4256a

January 2015

Applicable Products

HP Switch 2530-48G-PoE+	(J9772A)
HP Switch 2530-24G-PoE+	(J9773A)
HP Switch 2530-8G-PoE+	(J9774A)
HP Switch 2530-48G	(J9775A)
HP Switch 2530-24G	(J9776A)
HP Switch 2530-8G	(J9777A)
HP Switch 2530-48-PoE+	(J9778A)
HP Switch 2530-24-PoE+	(J9779A)
HP Switch 2530-8-PoE+	(J9780A)
HP Switch 2530-48	(J9781A)
HP Switch 2530-24	(J9782A)
HP Switch 2530-8	(J9783A)

Trademark Credits

Microsoft, Windows, and Microsoft Windows NT are US registered trademarks of Microsoft Corporation. Java™ is a US trademark of Sun Microsystems, Inc.

Disclaimer

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD COMPANY MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Warranty

See the Customer Support/Warranty booklet included with the product.

A copy of the specific warranty terms applicable to your Hewlett-Packard products and replacement parts can be obtained from your HP Sales and Service Office or authorized dealer.

Hewlett-Packard Company
8000 Foothills Boulevard, m/s 5551
Roseville, California 95747-5551
http://www.procurve.com

Contents

1 IPv6 Addressing Configuration

2 IPv6 Management Features

3 IPv6 Management Security Features

4 Multicast Listener Discovery (MLD) Snooping

IPv6 Addressing Configuration

Introduction

Feature	Default	CLI
Enable IPv6 with a Link-Local Address	disabled	1-4
Configure Global Unicast Autoconfig	disabled	1-5
Configure DHCPv6 Addressing	disabled	1-7
Configure a Static Link-Local Address	None	1-10
Configure a Static Global Unicast Address	None	1-11
Change DAD Attempts	3	1-16
View Current IPv6 Addressing	*n/a*	1-20

In the default configuration, IPv6 operation is disabled on the switch. This section describes the general steps and individual commands for enabling IPv6 operation.

This chapter provides the following:

- general steps for IPv6 configuration
- IPv6 command syntax descriptions, including **show** commands

Most IPv6 configuration commands are applied per-VLAN. The exceptions are ICMP, ND (neighbor discovery), and the (optional) authorized-managers feature, which are configured at the global configuration level. (ICMP and ND for IPv6 are enabled with default values when IPv6 is first enabled, and can either be left in their default settings or reconfigured, as needed.) For more information on ICMP, refer to “ICMP Rate-Limiting” on page 6-1.

Note The switch is capable of operating in dual-stack mode, where IPv4 and IPv6 run concurrently on a given VLAN.

General Configuration Steps

The IPv6 configuration includes global and per-VLAN settings. This section provides an overview of the general configuration steps for enabling IPv6 on a given VLAN and can be enabled by any one of several commands. The following steps provide a suggested progression for getting started.

Note The ICMP and Neighbor Discovery (ND) parameters are set to default values at the global configuration level are satisfactory for many applications and generally do not need adjustment when you are first configuring IPv6 on the switch.

In the default configuration, IPv6 is disabled on all VLANs.

1. If IPv6 DHCP service is available, enable IPv6 DHCP on the VLAN. If IPv6 is not already enabled on the VLAN, enabling DHCPv6 also enables IPv6 and automatically configures a link-local address using the EUI-64 format.

Note If IPv6 is not already enabled on the VLAN, enabling DHCPv6 causes the switch to automatically generate a link-local address. DHCPv6 does not assign a link-local address.

   A DHCPv6 server can provide other services, such as the addresses of time servers. For this reason you may want to enable DHCP even if you are using another method to configure IPv6 addressing on the VLAN.

2. If IPv6 DHCP service is not enabled on the VLAN, then do either of the following:
   - Enable IPv6 on the VLAN. This automatically configures a link-local address with an EUI- 64 interface identifier.
   - Statically configure a unicast IPv6 address on the VLAN. This enables IPv6 on the VLAN and, if you configure anything other than a link-local address, the link-local address will be automatically configured as well, with an EUI-64 interface identifier.

3. If an IPv6 router is connected on the VLAN, then enable IPv6 address autoconfiguration to automatically configure global unicast addresses with prefixes included in advertisements received from the router. The device identifier used in addresses configured by this method will be the same as the device identifier in the current link-local address.
4. If needed, statically configure IPv6 unicast addressing on the VLAN interface as needed. This can include any of the following:
   - statically replacing the automatically generated link-local address
   - statically adding global unicast, unique local unicast addresses

Configuring IPv6 Addressing

In the default configuration on a VLAN, any one of the following commands enables IPv6 and creates a link-local address. Thus, while any one of these methods is configured on a VLAN, IPv6 remains enabled and a link-local address is present:

ipv6 enable (page 1-4)

ipv6 address autoconfig (page 1-5)

ipv6 address dhcp full [rapid-commit] (page 1-7)

ipv6 address fe80:0:0:0:< *device-identifier* > link-local (page 1-10)

ipv6 address < *prefix:device-identifier* > (page 1-11)

Note Addresses created by any of these methods remain tentative until verified as unique by Duplicate Address Detection. (Refer to “Duplicate Address Detection (DAD)” on page 1-16.)

Enabling IPv6 with an Automatically Configured Link-Local Address

This command enables automatic configuration of a link-local address.

Syntax: [no] ipv6 enable

If IPv6 has not already been enabled on a VLAN by another IPv6 command option described in this chapter, this command enables IPv6 on the VLAN and automatically configures the VLAN's link-local unicast address with a 64-bit EUI-64 interface identifier generated from the VLAN MAC address.

> ***Note:*** *Only one link-local IPv6 address is allowed on the VLAN interface. Subsequent static or DHCP configuration of another link-local address overwrites the existing link-local address.*

A link-local address always uses the prefix fe80:0:0:0.

With IPv6 enabled, the VLAN uses received router advertisements to designate the default IPv6 router. (Refer to "Restricting IPv6 Router Advertisements" on page 1-27.)

After verification of uniqueness by DAD, a link-local IPv6 address assigned automatically is set to the **preferred** *status, with a "permanent" lifetime.*

Default: *Disabled*

The **no** *form of the command disables IPv6 on the VLAN if no other IPv6-enabling command is configured on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 1-13.)*

To view the current IPv6 Enable setting and any statically configured IPv6 addresses per-VLAN, use **show run**.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on the VLAN.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 1-20.

Enabling Automatic Configuration of a Global Unicast Address and a Default Router Identity on a VLAN

Enabling autoconfig or rebooting the switch with autoconfig enabled on a VLAN causes the switch to configure IPv6 addressing on the VLAN using router advertisements and an EUI-64 interface identifier.

Syntax: [no] ipv6 address autoconfig

Implements unicast address autoconfiguration as follows:

- *If IPv6 is not already enabled on the VLAN, this command enables IPv6 and generates a link-local (EUI- 64) address.*
- *Generates router solicitations (RS) on the VLAN.*
- *If a router advertisement (RA) is received on the VLAN, the switch uses the route prefix in the RA to configure a global unicast address. The device identifier for this address will be the same as the device identifier used in the current link-local address at the time the RA is received. (This can be either a statically configured or the (automatic) EUI-64 device identifier, depending on how the link-local address was configured.) If an RA is not received on the VLAN after autoconfig is enabled, a link-local address will be present, but no global unicast addresses will be autoconfigured.*

  Notes: *If a link-local address is already configured on the VLAN, a later, autoconfigured global unicast address uses the same device identifier as the link-local address.*

  Autoconfigured and DHCPv6-assigned global unicast addresses with the same prefix are mutually exclusive on a VLAN. On a given switch, if both options are configured on the same VLAN, then only the first to acquire a global unicast address will be used.

— Continued on the next page. —

— Continued from the previous page. —

After verification of uniqueness by DAD, an IPv6 address assigned to a VLAN by autoconfiguration is set to the preferred and valid lifetimes specified by the RA used to generate the address, and is configured as a preferred address.

Default: *Disabled.*

The **no** *form of the command produces different results, depending on how IPv6 is configured on the VLAN:*

If IPv6 was enabled only by the **autoconfig** *command, then deleting this command disables IPv6 on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 1-13.)*

To view the current IPv6 autoconfiguration settings per-VLAN, use **show run**.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on the VLAN.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 1-20.

Operating Notes

With IPv6 enabled, the VLAN uses received router advertisements to designate the default IPv6 router. (Refer to "Router Access and Default Router Selection" on page 1-26.)

Enabling DHCPv6

Enabling the DHCPv6 option on a VLAN allows the switch to obtain a global unicast address and an NTP (network time protocol) server assignment for a Timep server. (If a DHCPv6 server is not needed to provide a global unicast address to a switch interface, the server can still be configured to provide the NTP server assignment. This is sometimes referred to as "stateless DHCPv6".)

Syntax: [no] ipv6 address dhcp full [rapid-commit]

This option configures DHCPv6 on a VLAN, which initiates transmission of DHCPv6 requests for service. If IPv6 is not already enabled on the VLAN by the **ipv6 enable** *command, this option also enables IPv6 and causes the switch to autoconfigure a link-local unicast address with an EUI-64 interface identifier.*

> ***Notes:*** *A DHCPv6 server does not assign link-local addresses, and enabling DHCPv6 on a VLAN does not affect a pre-existing link-local address configured on the VLAN.*
>
> *A DHCPv6-assigned address can be configured on a VLAN when the following is true:*
>
> - *The assigned address is not on the same subnet as a previously configured autoconfig address.*
> - *The maximum IPv6 address limit on the VLAN or the switch has not been reached.*

If a DHCPv6 server responds with an IPv6 address assignment, this address is assigned to the VLAN. (The DHCPv6-assigned address will be dropped if it has the same subnet as another address already assigned to the VLAN by an earlier autoconfig command.)

— Continued on the next page. —

— Continued from the previous page. —

After verification of uniqueness by DAD, an IPv6 address assigned to the VLAN by an DHCPv6 server is set to the preferred and valid lifetimes specified in a router advertisement received on the VLAN for the prefix used in the assigned address, and is configured as a preferred address. (Refer to the section titled "Address Lifetimes" on page 1-34.)

[rapid-commit]: *Expedites DHCP configuration by using a two-message exchange with the server (solicit-reply) instead of the default four-message exchange (solicit-advertise- request-reply).*

Default: *Disabled*

The **no** *form of the command removes the DHCPv6 option from the configuration and, if no other IPv6-enabling command is configured on the VLAN, disables IPv6 on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 1-13.)*

To view the current IPv6 DHCPv6 settings per-VLAN, use **show run**.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on the VLAN.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 1-20.

Operating Notes

- If multiple DHCPv6 servers are available, the switch selects a server based on the preference value sent in DHCPv6 messages from the servers.
- The switch supports both DHCPv4 and DHCPv6 client operation on the same VLAN.
- With IPv6 enabled, the switch determines the default IPv6 router for the VLAN from the router advertisements it receives. (Refer to "Restricting IPv6 Router Advertisements" on page 1-27.)

- DHCPv6 and statically configured global unicast addresses are mutually exclusive on a given VLAN. That is, configuring DHCPv6 on a VLAN erases any static global unicast addresses previously configured on that VLAN, and the reverse. (A statically configured link-local address will not be affected by configuring DHCPv6 on the VLAN.)
- For the same subnet on the switch, a DHCPv6 global unicast address assignment takes precedence over an autoconfigured address assignment, regardless of which address type was the first to be configured. If DHCPv6 is subsequently removed from the configuration, then an autoconfigured address assignment will replace it after the next router advertisement is received on the VLAN. DHCPv6 and autoconfigured addresses co-exist on the same VLAN if they belong to different subnets.

For related information refer to:

- RFC 3315: “Dynamic Host Configuration Protocol for IPv6 (DHCPv6)”
- RFC 3633: “IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6”
- RFC 3736: “Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6”

Configuring a Static IPv6 Address on a VLAN

This option enables configuring of unique, static unicast IPv6 addresses for global and link-local applications, including:

- link-local unicast (including EUI and non-EUI device identifiers)
- global unicast (and unique local unicast)

Statically Configuring a Link-Local Unicast Address

Syntax: [no] ipv6 address fe80::< *device-identifier* > link-local

- *If IPv6 is not already enabled on the VLAN, this command enables IPv6 and configures a static link-local address.*
- *If IPv6 is already enabled on the VLAN, then this command overwrites the current, link- local address with the specified static address. (One link-local address is allowed per VLAN interface.)*

< *device-identifier* >: The low-order 64 bits, in 16-bit blocks, comprise this value in a link-local address:

xxxx xxxx : xxxx xxxx : xxxx xxxx : xxxx xxxx

Where a static link-local address is already configured, a new, autoconfigured global unicast addresses assignment uses the same device identifier as the link-local address.

> ***Notes:*** *An existing link-local address is replaced, and is not deprecated, when a static replacement is configured.*
>
> *The prefix for a statically configured link-local address is always 64 bits, with all blocks after fe80 set to zero. That is: fe80:0:0:0.*

After verification of uniqueness by DAD, a statically configured link-local address status is set to **preferred**, *with a* **permanent** *lifetime.*

For link-local addressing, the ***no*** *form of the static IPv6 address command produces different results, depending on how IPv6 is configured on the VLAN:*

- *If IPv6 was enabled only by a statically configured link-local address, then deleting the link-local address disables IPv6 on the VLAN.*
- *If other IPv6-enabling commands have been configured on the VLAN, then deleting the statically configured link-local address causes the switch to replace it with the default (EUI-64) link-local address for the VLAN, and IPv6 remains enabled.*

Refer also to "Disabling IPv6 on a VLAN" on page 1-13.

Statically Configuring A Global Unicast Address

Syntax:. [no] ipv6 address < network-prefix><device-id >/< prefix-length >
[no] ipv6 address < network-prefix>::/< prefix-length > eui-64

If IPv6 is not already enabled on a VLAN, either of these command options do the following:

- *enable IPv6 on the VLAN*
- *configure a link-local address using the EUI-64 format*
- *statically configure a global unicast address*

If IPv6 is already enabled on the VLAN, then the above commands statically configure a global unicast address, but have no effect on the current link-local address.

< *network-prefix* >: *This includes the global routing prefix and the subnet ID for the address.*

< *device-id* >: Enters a user-defined device identity.

< *prefix-length* >: Specifies the number of bits in the network prefix. If you are using the **eui-64** option, this value must be 64.

eui-64: *Specifies using the Extended Unique Identifier format to create a device identifier based on the VLAN MAC address.*

After verification of uniqueness by DAD, the lifetime of a statically configured IPv6 address assigned to a VLAN is set to permanent, and is configured as a preferred address.

The **no** *form of the command erases the specified address and, if no other IPv6-enabling command is configured on the VLAN, disables IPv6 on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 1-13.)*

To view the currently configured static IPv6 addresses per-VLAN, use **show run**.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on **VLAN < *vid* >**.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 1-20.

Operating Notes

- With IPv6 enabled, the switch determines the default IPv6 router for the VLAN from the router advertisements it receives. (Refer to "Router Access and Default Router Selection" on page 1-26.)
- If DHCPv6 is configured on a VLAN, then configuring a static global unicast address on the VLAN removes DHCPv6 from the VLAN's configuration and deletes the DHCPv6-assigned global unicast address.
- Note that for a statically configured global unicast address to be routable, a gateway router must be transmitting router advertisements on the VLAN.
- If an autoconfigured global unicast address already exists for the same subnet as a new, statically configured global unicast address, the statically configured address is denied. In the reverse case, you can add an autoconfig command to the VLAN configuration, but it will not be implemented unless the static address is removed from the configuration.

Syntax:. [no] ipv6 address < network-prefix >< device-identifier >/< prefix-length >

If IPv6 is not already enabled on a VLAN, this command option does the following:

- *enables IPv6 on the VLAN*
- *configures a link-local address using the EUI-64 format*

Default: *None.*

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on **VLAN < *vid* >**.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 1-20.

Duplicate Address Detection (DAD) for Statically Configured Addresses

Statically configured IPv6 addresses are designated as permanent. If DAD determines that a statically configured address duplicates a previously configured and reachable address on another device belonging to the VLAN, then the more recent, duplicate address is designated as **duplicate**. For more on this topic, refer to:

- "Duplicate Address Detection (DAD)" on page 1-16.
- "View the Current IPv6 Addressing Configuration" on page 1-20

Disabling IPv6 on a VLAN

While one IPv6-enabling command is configured on a VLAN, IPv6 remains enabled on that VLAN. In this case, removing the only IPv6-enabling command from the configuration disables IPv6 operation on the VLAN. That is, to disable IPv6 on a VLAN, all of the following commands must be removed from the VLAN's configuration:

```
ipv6 enable

ipv6 address dhcp full [rapid-commit]
```

ipv6 address autoconfig

ipv6 address fe80::< *device-identifier* > link-local

ipv6 address < *prefix* > : < *device-identifier* >

If any of the above remain enabled, then IPv6 remains enabled on the VLAN and, at a minimum, a link-local unicast address will be present.

Neighbor Discovery (ND)

Neighbor Discovery (ND) is the IPv6 equivalent of the IPv4 ARP for layer 2 address resolution, and uses IPv6 ICMP messages to do the following:

- Determine the link-layer address of neighbors on the same VLAN interface.
- Verify that a neighbor is reachable.
- Track neighbor (local) routers.

Neighbor Discovery enables functions such as the following:

- router and neighbor solicitation and discovery
- detecting address changes for devices on a VLAN
- identifying a replacement for a router or router path that has become unavailable
- duplicate address detection (DAD)
- router advertisement processing
- neighbor reachability
- autoconfiguration of unicast addresses
- resolution of destination addresses
- changes to link-layer addresses

An instance of Neighbor Discovery is triggered on a device when a new (tentative) or changed IPv6 address is detected. (This includes stateless, stateful, and static address configuration.) ND operates in a per-VLAN scope; that is, within the VLAN on which the device running the ND instance is a member. Neighbor discovery actually occurs when there is communication between devices on a VLAN. That is, a device needing to determine the link-layer address of another device on the VLAN initiates a (multicast) neighbor solicitation message (containing a solicited-node multicast address that corresponds to the IPv6 address of the destination device) on the VLAN. When the destination device receives the neighbor solicitation, it responds with a neighbor advertisement message identifying its link-layer address. When the initiating device receives this advertisement, the two devices are ready to exchange traffic on the VLAN interface. Also, when an IPv6 interface becomes operational, it transmits a router solicitation on the interface and listens for a router advertisement.

Note

Neighbor and router solicitations must originate on the same VLAN as the receiving device. To support this operation, IPv6 is designed to discard any incoming neighbor or router solicitation that does not have a value of 255 in the IP Hop Limit field. For a complete list of requirements, refer to RFC 246.

When a pair of IPv6 devices in a VLAN exchange communication, they enter each other's IPv6 and corresponding MAC addresses in their respective neighbor caches. These entries are maintained for a period of time after communication ceases, and then dropped.

To view or clear the content of the neighbor cache, refer to “Viewing and Clearing the IPv6 Neighbors Cache” on page 2-1.

For related information, refer to:

- RFC 2461: “Neighbor Discovery for IP Version 6 (IPv6)”

Duplicate Address Detection (DAD)

Duplicate Address Detection verifies that a configured unicast IPv6 address is unique before it is assigned to a VLAN interface on the switch. DAD is enabled in the default IPv6 configuration, and can be reconfigured, disabled, or re-enabled at the global config command level. DAD can be useful in helping to troubleshoot erroneous replies to DAD requests, or where the neighbor cache contains a large number of invalid entries due to an unauthorized station sending false replies to the switch's neighbor discovery queries. If DAD verifies that a unicast IPv6 address is a duplicate, the address is not used. If the link-local address of the VLAN interface is found to be a duplicate of an address for another device on the interface, then the interface stops processing IPv6 traffic.

DAD Operation

On a given VLAN interface, when a new unicast address is configured, the switch runs DAD for this address by sending a neighbor solicitation to the All-Nodes multicast address (ff02::1). This operation discovers other devices on the VLAN and verifies whether the proposed unicast address assignment is unique on the VLAN. (During this time, the address being checked for uniqueness is held in a tentative state, and cannot be used to receive traffic other than neighbor solicitations and neighbor advertisements.) A device that receives the neighbor solicitation responds with a Neighbor Advertisement

that includes its link-local address. If the newly configured address is from a static or DHCPv6 source and is found to be a duplicate, it is labelled as duplicate in the "Address Status" field of the **show ipv6** command, and is not used. If an autoconfigured address is found to be a duplicate, it is dropped and the following message appears in the Event Log:

```
W < date > < time > 00019 ip: ip address < IPv6-address >
removed from vlan id < vid >
```

DAD does not perform periodic checks of existing addresses. However, when a VLAN comes up with IPv6 unicast addresses configured (as can occur during a reboot) the switch runs DAD for each address on the interface by sending neighbor solicitations to the All-Nodes multicast address as described above.

If an address is configured while DAD is disabled, the address is assumed to be unique and is assigned to the interface. If you want to verify the uniqueness of an address configured while DAD was disabled, re-enable DAD and then either delete and reconfigure the address, or reboot the switch.

Configuring DAD

Syntax: ipv6 nd dad-attempts < 0 - 600 >

This command is executed at the global config level, and configures the number of neighbor solicitations to send when performing duplicate address detection for a unicast address configured on a VLAN interface.

< 0 - 600 >: *The number of consecutive neighbor solicitation messages sent for DAD inquiries on an interface. Setting this value to 0 disables DAD on the interface. Disabling DAD bypasses checks for uniqueness on newly configured addresses. If a reboot is performed while DAD is disabled, the duplicate address check is not performed on any IPv6 addresses configured on the switch.*

Default: *3 (enabled); Range: 0 - 600 (0 = disabled)*

The **no** form of the command restores the default setting (3).

Syntax: ipv6 nd ns-interval < *milliseconds* >

Used on VLAN interfaces to reconfigure the neighbor discovery time in milliseconds between DAD neighbor solicitations sent for an unresolved destination, or between duplicate address detection neighbor solicitation requests. Increasing this setting is indicated where neighbor solicitation retries or failures are occurring, or in a "slow" (WAN) network.

This value can be configured in a router advertisement to help ensure that all hosts on a VLAN are using the same retransmit interval for neighbor discovery. Refer to "Set or Change the Neighbor Discovery Retransmit Timer" on page 8-13.

To view the current setting, use **show ipv6 nd**.

Range: *1000 - 3600000 ms;* ***Default:*** *1000 ms.*

Syntax:. ipv6 nd reachable-time < *milliseconds* >

Used on VLAN interfaces to configure the length of time in milliseconds a neighbor will be considered reachable after the Neighbor Unreachability Detection algorithm has confirmed it to be reachable. When the switch operates in host mode, this setting can be overridden by a reachable time received in a router advertisement.

This value can be configured in a router advertisement to help ensure that all hosts on a VLAN are using the same reachable time in their neighbor cache. Refer to "Change the Reachable Time Duration for Neighbors" on page 8-12.

To view the current setting, use **show ipv6 nd**.

Range: *1000 - 2147483647 ms;* ***Default:*** *30000 ms.*

Operating Notes

- A verified link-local unicast address must exist on a VLAN interface before the switch can run DAD on other addresses associated with the interface.
- If a previously configured unicast address is changed, a neighbor advertisement (an all-nodes multicast message–ff02::1) is sent to notify other devices on the VLAN and to perform duplicate address detection.
- IPv6 addresses on a VLAN interface are assigned to multicast address groups identified with well- known prefixes.
- DAD is performed on all stateful, stateless, and statically configured unicast addresses.
- Neighbor solicitations for DAD do not cause the neighbor cache of neighboring switches to be updated.
- If a previously configured unicast address is changed, a neighbor advertisement is sent on the VLAN to notify other devices, and also for duplicate address detection.
- If DAD is disabled when an address is configured, the address is assumed to be unique and is assigned to the interface.

View the Current IPv6 Addressing Configuration

Use these commands to view the current status of the IPv6 configuration on the switch.

Syntax: show ipv6

Lists the current, global IPv6 settings and per-VLAN IPv6 addressing on the switch.

IPv6 Routing: *This setting is always* **Disabled**. *This is a global setting, and is not configured per-VLAN. (Refer to "Router Access and Default Router Selection" on page 1-26.)*

Default Gateway: *Lists the IPv4 default gateway, if any, configured on the switch. This is a globally configured router gateway address, and is not configured per-VLAN.*

ND DAD: *Indicates whether DAD is enabled (the default) or disabled. Using* **ipv6 nd dad-attempts 0** *disables neighbor discovery. (Refer to "Duplicate Address Detection (DAD)" on page 1-16.)*

DAD Attempts: *Indicates the number of neighbor solicitations the switch transmits per-address for duplicate (IPv6) address detection. Implemented when a new address is configured or when an interface with configured addresses comes up (such as after a reboot). The default setting is 3, and the range is 0 - 600. A setting of "0" disables duplicate address detection. (Refer to "Duplicate Address Detection (DAD)" on page 1-16.)*

VLAN Name: *Lists the name of a VLAN statically configured on the switch.*

IPv6 Status: *For the indicated VLAN, indicates whether IPv6 is disabled (the default) or enabled. (Refer to "Configuring IPv6 Addressing" on page 1-3.)*

Address Origin:

- **Autoconfig:** *The address was configured using stateless address autoconfiguration (SLAAC). In this case, the device identifier for global unicast addresses copied from the current link-local unicast address.*
- **DHCP:** *The address was assigned by a DHCPv6 server. Note that addresses having a DHCP origin are listed with a 128-bit prefix length.*
- **Manual:** *The address was statically configured on the VLAN.*
- **IPv6 Address/Prefix Length:** *Lists each IPv6 address and prefix length configured on the indicated VLAN.*

Address Status:

- **Tentative:** *DAD has not yet confirmed the address as unique, and is not usable for sending and receiving traffic.*
- **Preferred:** *The address has been confirmed as unique by DAD, and usable for sending and receiving traffic. The Expiry time shown for this address by the* **show ipv6 vlan <** ***vid*****>** *command output is the preferred lifetime assigned to the address. (Refer to "Address Lifetimes" on page xxx.)*
- **Deprecated:** *The preferred lifetime for the address has been exceeded, but there is time remaining in the valid lifetime.*
- **Duplicate:** *Indicates a statically configured IPv6 address that is a duplicate of another IPv6 address that already exists on another device belonging to the same VLAN interface. A duplicate address is not used.*

For example, figure 1-1 shows the output on a switch having IPv6 enabled on one VLAN.

```
HP Switch(config)# show ipv6

 Internet (IPv6) Service

  IPv6 Routing    : Disabled
  Default Gateway : 10.0.9.80
  ND DAD          : Enabled
  DAD Attempts    : 3

  Vlan Name       : DEFAULT_VLAN
  IPv6 Status     : Disabled

  Vlan Name       : VLAN10
  IPv6 Status     : Enabled

  Address    |                                          Address
  Origin     | IPv6 Address/Prefix Length               Status
  ---------- + ---------------------------------------- -----------
  autoconfig | 2620:0:a03:e102::127/64                  preferred
  dhcp       | 2620:0:a03:e102:212:79ff:fe88:a100/64    preferred
  manual     | fe80::127/64                             preferred
```

Figure 1-1. Example of Show IPv6 Command Output

Syntax: show ipv6 vlan < *vid* >

Displays IP and IPv6 global configuration settings, the IPv6 status for the specified VLAN, the IPv6 addresses (with prefix lengths) configured on the specified VLAN, and the expiration data (Expiry) for each address.:

- **IPv6 Routing:** T*his setting is always* **Disabled**. *(Refer to "Router Access and Default Router Selection" on page 1-26.).*
- **Default Gateway:** *Lists the IPv4 default gateway, if any, configured on the switch. This is a globally configured router gateway address, and is not configured per-VLAN.*
- **ND DAD:** *Shows whether Neighbor Discovery (ND) is enabled. The default setting is Enabled. Using* **ipv6 nd dad-attempts 0** *disables neighbor discovery.*

- **DAD Attempts:** *Indicates the number of neighbor solicitations the switch transmits per-address for duplicate (IPv6) address detection. Implemented when a new address is configured or when an interface with configured addresses comes up (such as after a reboot). The default setting is 3, and the range is 0 - 600. A setting of "0" disables duplicate address detection. (Refer to "Duplicate Address Detection (DAD)" on page 1-16.)*
- **VLAN Name:** *Lists the name of a VLAN statically configured on the switch.*
- **IPv6 Status:** *For the indicated VLAN, indicates whether IPv6 is disabled (the default) or enabled. (Refer to "Configuring IPv6 Addressing" on page 1-3.)*
- **IPv6 Address/Prefix Length:** *Lists each IPv6 address and prefix length configured on the indicated VLAN.*
- **Expiry:** *Lists the lifetime status of each IPv6 address listed for a VLAN:*
  - **Permanent:** *The address will not time out and need renewal or replacement.*
  - **date/time:** *The date and time that the address expires. Expiration date and time is specified in the router advertisement used to create the prefix for automatically configured, global unicast addresses. The* **Address Status** *field in the* **show ipv6** *command output indicates whether this date/time is for the "preferred" or "valid" lifetime assigned to the corresponding address.*

```
HP Switch(config)# show ipv6 vlan 10

 Internet (IPv6) Service

  IPv6 Routing    : Disabled
  Default Gateway : 10.0.9.80
  ND DAD          : Enabled
  DAD Attempts    : 3

  Vlan Name       : VLAN10
  IPv6 Status     : Enabled

  IPv6 Address/Prefixlength                   Expiry
  ------------------------------------------- ------------------------
  2620:0:a03:e102::127/64                     Wed Jan 23 14:16:17 2008

  2620:0:a03:e102:212:79ff:fe88:a100/64       Sat Jan  5 05:02:22 2008

  fe80::127/64                                permanent
```

Figure 1-2. Example of Show IPv6 VLAN < *vid* > Output

Syntax: show run

In addition to the other elements of the current configuration, this command lists the statically configured, global unicast IPv6 addressing, and the current IPv6 configuration per-VLAN. The listing may include one or more of the following, depending on what other IPv6 options are configured on the VLAN. Any stateless address autoconfiguration (SLAAC) commands in the configuration are also listed in the output, but the actual addresses resulting from these commands are not included in the output.

- ipv6 enable
- ipv6 address fe80::< *device-id* > link-local
- ipv6 address < *prefix* >:< *device-id* >/< *prefix-length* >
- ipv6 address autoconfig
- ipv6 address dhcp full [rapid-commit]
- ipv6 < *global-unicast-address* >/< *prefix* >

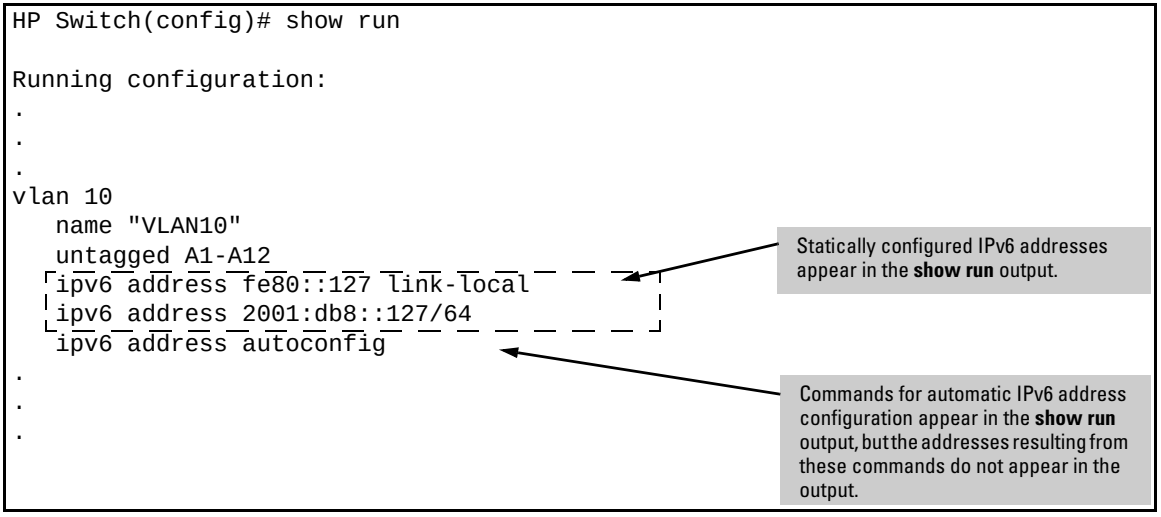


Figure 1-3. Example of Show Run Output Listing the Current IPv6 Addressing Commands

Router Access and Default Router Selection

Routing traffic between destinations on different VLANs configured on the switch or to a destination on an off-switch VLAN is done by placing the switch on the same VLAN interface or subnet as an IPv6-capable router configured to route traffic to other IPv6 interfaces.

Router Advertisements

An IPv6 router periodically transmits router advertisements (RAs) on the VLANs to which it belongs to notify other devices of its presence. The switch uses these advertisements for purposes such as:

- learning the MAC and link-local addresses of IPv6 routers on the VLAN (For devices other than routers, the switch must use neighbor discovery to learn these addresses.)
- building a list of default (reachable) routers, along with router lifetime and prefix lifetime data
- learning the prefixes and the valid and preferred lifetimes to use for stateless (autoconfigured) global unicast addresses (This is required for autoconfiguration of global unicast IPv6 addresses.)
- learning the hop limit for traffic leaving the VLAN interface
- learning the MTU (Maximum Transmission Unit) to apply to frames intended to be routed

Router Solicitations

When an IPv6 interface becomes operational on the switch, a router solicitation is automatically sent to trigger a router advertisement (RA) from any IPv6 routers reachable on the VLAN. (Router solicitations are sent to the All-Routers multicast address; ff02::2. If an RA is not received within one second of sending the initial router solicitation, the switch sends up to three additional solicitations at intervals of four seconds. If an RA is received, the sending router is added to the switch's default router list and the switch stops sending router solicitations. If an RA is not received, then IPv6 traffic on that VLAN cannot be routed, and the only usable unicast IPv6 address on the VLAN is the link-local address.

Note If the switch does not receive a router advertisement after sending the router solicitations, as described above, then no further router solicitations are sent on that VLAN unless a new IPv6 setting is configured, IPv6 on the VLAN is disabled, then re-enabled, or the VLAN itself is disconnected, then reconnected.

Restricting IPv6 Router Advertisements

The RA Guard feature restricts the ports (or trunks) that can accept IPv6 Router Advertisements (RAs). Additionally, ICMPv6 router redirects are blocked on the configured ports.

Only physical ports and trunk ports are supported. Dynamic ports, dynamic trunks, and mesh ports are not supported.

Note IPv6 RAs are ICMPv6 type 134 messages and may be sent to either the "all nodes" multicast address (FF02:0:0:0:0:0:0:1) or to the address of the device itself as a result of an IPv6 router solicitation. IPv6 router redirect messages are ICMPv6 type 137 messages. They are sent to the source address of the packet that triggered the redirect.

Configuring RA Guard

Syntax: [no] ipv6 ra-guard ports <*port-list*> [log]

Enables or disable RA Guard on the specified ports, which blocks IPv6 router advertisements and router redirects.

The **no** *form of the command disables RA Guard.*

[log]: *Enables debug logging of RA and redirects packets to debug output.*

```
HP Switch(config)# ipv6 ra-guard ports 6 log
```

Figure 1-4. Enabling RA Guard

Operating Notes

- When a logical trunk port is enabled, all members of the trunk are enabled for RA Guard. Likewise, when a logical trunk port is disabled, (**no ipv6 ra-guard ports <*trunk-port*>**), all members of the trunk are disabled for RA Guard.
- When ports are configured for RA Guard, hardware resources are allocated. If there are not enough hardware resources, this message displays:

  ```
  Commit failed
  ```

- When debug logging is enabled (**ipv6 ra-guard ports <*port-list*> log**), the RA and redirect packets are sent to the CPU, which can be CPU-intensive. This message displays:

  ```
  The log option uses a lot of CPU and should be used
  only for short periods of time.
  ```

- The **debug security ra-guard** command is used to filter and display RA Guard debug log messages.

To display configuration and statistical information about RA Guard, enter the **show ipv6 ra-guard** command.

```
HP Switch(config)# show ipv6 ra-guard

 IPv6 RA Guard Information

  Port  Block RAs Blocked Redirs Blocked Log
  ----- ----- ----------- -------------- ---
  1     No    0           0              No
  2     No    0           0              No
  3     No    0           0              No
  4     No    0           0              No
  5     No    0           0              No
  6     Yes   123         450            Yes
  7     No    0           0              No
  8     No    0           0              No
```

Figure 1-5. Output Showing Configuration and Statistics for RA Guard

When RA Guard is enabled, there will be one or two lines displayed in the running config file.

```
HP Switch(config)# show running-config

Running configuration:

; J8693A Configuration Editor; Created on release #K.15.07.0000x
; Ver #02:01.0f:0c

hostname "HP Switch"
module 1 type J86yyA
module 2 type J86xxA
module 3 type J8694A
no stack auto-join
vlan 1
   name "DEFAULT_VLAN"
   untagged 1-4,7-48,A1-A4
   ipv6 address fe80::2 link-local
   ip address dhcp-bootp
   ipv6 enable
   no untagged 5-6
   exit
vlan 2
   name "VLAN2"
   untagged 5-6
   ip address 10.10.10.1 255.255.255.0
   exit
power-over-ethernet pre-std-detect
sflow 3 destination 3fff::3
ipv6 unicast-routing
ipv6 ra-guard ports 6 log   <-- RA Guard is enabled on port 6; logging is enabled.
...
```

Figure 1-6. Running Config File Showing Line for RA-Guard

Default IPv6 Router

If IPv6 is enabled on a VLAN where there is at least one accessible IPv6 router, the switch selects a default IPv6 router. (Refer to "Enabling Automatic Configuration of a Global Unicast Address and a Default Router Identity on a VLAN" on page 1-5.)

- If the switch receives router advertisements (RAs) from a single IPv6 router on the same VLAN or subnet, the switch configures a global unicast address and selects the advertising router as the default IPv6 router.
- If multiple IPv6 routers on a VLAN send RAs advertising the same network, the switch configures one global unicast address and selects one router as the default router, based on the router's relative reachability, using factors such as router priority and route cost.

- If multiple IPv6 routers on a VLAN send RAs advertising different subnets, the switch configures a corresponding global unicast address for each RA and selects one of the routers as the default IPv6 router, based on route cost. When multiple RAs are received on a VLAN, the switch uses the router priority and route cost information included in the RAs to identify the default router for the VLAN.

Router Redirection

With multiple routers on a VLAN, if the default (first-hop) router for an IPv6-enabled VLAN on the switch determines that there is a better first-hop router for reaching a given, remote destination, the default router can redirect the switch to use that other router as the default router. For further information on routing IPv6 traffic, refer to the documentation provided for the IPv6 router.

For related information:

- RFC 2461: “Neighbor Discovery for IP Version 6”

View IPv6 Gateway, Route, and Router Neighbors

Use these commands to view the switch's current routing table content and connectivity to routers per VLAN. This includes information received in router advertisements from IPv6 routers on VLANs enabled with IPv6 on the switch.

Viewing Gateway and IPv6 Route Information

Syntax: show ipv6 route [*ipv6-addr*] [connected]

This command displays the routes in the switch's IPv6 routing table.

ipv6-addr*. Optional. Limits the output to show the gateway to the specified IPv6 address.*

connected: *Optional. Limits the output to show only the gateways to IPv6 addresses connected to VLAN interfaces configured on the switch, including the loopback (::1/128) address.*

Dest: *The destination address for a detected route.*

Gateway: *The IPv6 address or VLAN interface used to reach the destination. (Includes the loopback address.)*

Type: *Indicates route type (static, connected, RIP, or OSPF).*

Distance: *The route's administrative distance, used to determine the best path to the destination.*

Metric: *Indicates the route cost for the selected destination.*

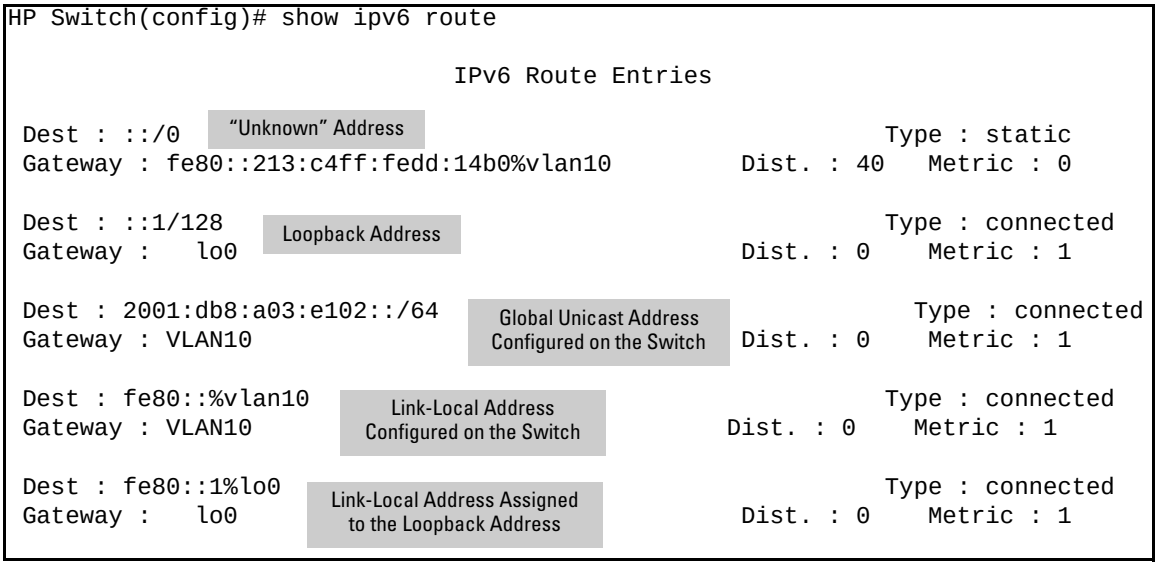


Figure 1-7. Example of Show IPv6 Route Output

Viewing IPv6 Router Information

Syntax: show ipv6 routers [vlan < *vid* >]

This command lists the switch's IPv6 router table entries for all VLANs configured on the switch or for a single VLAN. This output provides information about the IPv6 routers from which routing advertisements (RAs) have been received on the switch.

vlan < *vid* >: Optional. Specifies only the information on IPv6 routers on the indicated VLAN.

Router Address: *The IPv6 address of the router interface.*

Preference: *The relative priority of prefix assignments received from the router when prefix assignments are also received on the same switch VLAN interface from other IPv6 routers.*

Interface: *The VLAN interface on which the path to the router exists.*

MTU: *This is the Maximum Transmission Unit (in bytes) allowed for frames on the path to the indicated router.*

Hop Limit: *The maximum number of router hops allowed.*

Prefix Advertised: *Lists the prefix and prefix size (number of leftmost bits in an address) originating with the indicated router.*

Valid Lifetime: *The total time the address is available, including the preferred lifetime and the additional time (if any) allowed for the address to exist in the deprecated state. Refer to "Address Lifetimes" on page 1-34.*

Preferred Lifetime: *The length of time during which the address can be used freely as both a source and a destination address for traffic exchanges with other devices. Refer to "Address Lifetimes" on page 1-34.*

On/Off Link: Indicates whether the entry source is on the same VLAN as is indicated in the **Interface** field.

For example, figure 1-8 indicates that the switch is receiving router advertisements from a single router that exists on VLAN 10.

```
HP Switch(config)# show ipv6 routers

 IPv6 Router Table Entries

  Router Address : fe80::213:c4ff:fedd:14b0
  Preference     : Medium
  Interface      : VLAN10
  MTU            : 1500
  Hop Limit      : 64

                                              Valid        Preferred    On/Off
  Prefix Advertised                           Lifetime(s)  Lifetime(s)  Link
  ------------------------------------------- ------------ ------------ -------
  2001:db8:a03:e102::/64                     864000       604800       Onlink
```

Figure 1-8. Example of Show IPv6 Routers Output

Address Lifetimes

Every configured IPv6 unicast address has a lifetime setting that determines how long the address can be used before it must be refreshed or replaced. Some addresses are set as “permanent” and do not expire. Others have both a “preferred” and a “valid” lifetime that specify the duration of their use and availability.

Preferred Lifetime

This is the length of time during which the address can be used freely as both a source and a destination address for traffic exchanges with other devices. This time span is equal to or less than the valid lifetime also assigned to the address. If this time expires without the address being refreshed, the address becomes deprecated and should be replaced with a new, preferred address. In the deprecated state, an address can continue to be used as a destination for existing communication exchanges, but is not used for new exchanges or as a source for traffic sent from the interface. A new, preferred address and its deprecated counterpart will both appear in the **show ipv6 vlan < *vid* >** output as long as the deprecated address is within its valid lifetime.

Valid Lifetime

This is the total time the address is available, and is equal to or greater than the preferred lifetime. The valid lifetime enables communication to continue for transactions that began before the address became deprecated. However, in this timeframe, the address should no longer be used for new communications. If this time expires without the deprecated address being refreshed, the address becomes invalid and may be assigned to another interface.

Sources of IPv6 Address Lifetimes

Manually configured addresses have permanent lifetimes. The prefixes received from router advertisements for global unicast addresses include finite valid and preferred lifetime assignments.

Table 1-1. IPv6 Unicast Addresses Lifetimes

Address Source	Lifetime Criteria
Link-Local	Permanent
Statically Configured Unicast	Permanent
Autoconfigured Global	Finite Preferred and Valid Lifetimes
DHCPv6-Configured	Finite Preferred and Valid Lifetimes

A new, preferred address used as a replacement for a deprecated address can be acquired from a manual, DHCPv6, or autoconfiguration source.

IPv6 Management Features

Introduction

Feature	Default	CLI
Neighbor Cache	n/a	2-2, 2-4
Telnet6	Enabled	2-5, 2-6, 2-7
SNTP Address	None	2-9
Timep Address	None	2-12
TFTP	n/a	2-14
SNMP Trap Receivers	None	2-21

This chapter focuses on the IPv6 application of management features that support both IPv6 and IPv4 operation. For additional information on these features, refer to the current *Management and Configuration Guide* for your switch.

Viewing and Clearing the IPv6 Neighbors Cache

Neighbor discovery occurs when there is communication between the switch and another, reachable IPv6 device on the same VLAN. A neighbor destination is reachable from a given source address if a confirmation (neighbor solicitation) has been received at the source verifying that traffic has been received at the destination.

The switch maintains an IPv6 neighbor cache that is populated as a result of communication with other devices on the same VLAN. You can view and clear the contents of the neighbor cache using the commands described in this section.

Viewing the Neighbor Cache

Neighbor discovery occurs when there is communication between IPv6 devices on a VLAN. The Neighbor Cache retains data for a given neighbor until the entry times out. For more on this topic, refer to "Neighbor Discovery (ND)" on page 1-15.

Syntax: show ipv6 neighbors [vlan < *vid* >]

Displays IPv6 neighbor information currently held in the neighbor cache. After a period without communication with a given neighbor, the switch drops that neighbor's data from the cache. The command lists neighbors for all VLAN interfaces on the switch or for only the specified VLAN. The following fields are included for each entry in the cache:

IPv6 Address: *Lists the 128-bit addresses for the local host and any neighbors (on the same VLAN) with whom there has been recent communication.*

MAC Address: *The MAC Address corresponding to each of the listed IPv6 addresses.*

VLAN < *vid* >: *Optional. Causes the switch to list only the IPv6 neighbors on a specific VLAN configured on the switch.*

Type: *Appears only when VLAN is not specified, and indicates whether the corresponding address is* ***local*** *(configured on the switch) or* **dynamic** *(configured on a neighbor device).*

Age: *Appears only when VLAN is specified, and indicates the length of time the entry has remained unused.*

Port: *Identifies the switch port on which the entry was learned. If this field is empty for a given address, then the address is configured on the switch itself.*

State: *A neighbor destination is reachable from a given source address if confirmation has been received at the source verifying that traffic has been received at the destination. This field shows the reachability status of each listed address:*

- **INCOM** (*Incomplete*)**:** *Neighbor address resolution is in progress, but has not yet been determined.*
- **REACH** (*Reachable*)**:** *The neighbor is known to have been reachable recently.*

— Continued on the next page. —

— Continued from previous page. —

- **STALE:** *A timeout has occurred for reachability of the neighbor, and an unsolicited discovery packet has been received from the neighbor address. If the path to the neighbor is then used successfully, this state is restored to* **REACH**.
- **DELAY:** *Indicates waiting for a response to traffic sent recently to the neighbor address. The time period for determining the neighbor's reachability has been extended.*
- **PROBE:** *The neighbor may not be reachable. Periodic, unicast neighbor solicitations are being sent to verify reachability.*

```
HP Switch(config)# show ipv6 neighbor

 IPv6 ND Cache Entries

 IPv6 Address                              MAC Address   State Type    Port
 ----------------------------------------- ------------- ----- ------- ----
 2001:db8:260:212::101                     0013c4-dd14b0 STALE dynamic A1
 2001:db8:260:214::1:15                    001279-88a100 REACH local
 fe80::1:1                                 001279-88a100 REACH local
 fe80::10:27                               001560-7aadc0 REACH dynamic A3
 fe80::213:c4ff:fedd:14b0                  0013c4-dd14b0 REACH dynamic A1
```

Figure 2-1. Example of Neighbor Cache Without Specifying a VLAN

```
HP Switch(config)# show ipv6 neighbor vlan 10

 IPv6 ND Cache Entries

 IPv6 Address                              MAC Address   State Age           Port
 ----------------------------------------- ------------- ----- ------------- ----
 2001:db8:260:212::101                     0013c4-dd14b0 STALE 5h:13m:44s    A1
 2001:db8:260:214::1:15                    001279-88a100 REACH 11h:15m:23s   B17
 fe80:1a3::1:1                             001279-88a100 REACH 9h:35m:11s    B12
 fe80:::10:27                              001560-7aadc0 REACH 22h:26m:12s   A3
 fe80::213:c4ff:fedd:14b0                  0013c4-dd14b0 REACH 23 0h:32m:36s A1
```

Figure 2-2. Example of Neighbor Cache Content for a Specific VLAN

Clearing the Neighbor Cache

When there is an event such as a topology change or an address change, the neighbor cache may have too many entries to allow efficient use. Also, if an unauthorized client is answering DAD or normal neighbor solicitations with invalid replies, the neighbor cache may contain a large number of invalid entries and communication with some valid hosts may fail and/or the **show ipv6 neighbors** command output may become too cluttered to efficiently read. In such cases, the fastest way to restore optimum traffic movement on a VLAN may be to statically clear the neighbor table instead of waiting for the unwanted entries to time-out.

Syntax: clear ipv6 neighbors

Executed at the global config level, this command removes all non-local IPv6 neighbor addresses and corresponding MAC addresses from the neighbor cache. (Local IPv6 addresses, that is, IPv6 addresses configured on the VLAN interface for the switch on which the command is executed, are not removed.) Removed addresses are listed in the command output.

```
HP Switch(config)# clear ipv6 neighbors

2001:db8:260:212::1%vlan10 deleted
fe80:::10:27%vlan10 deleted
fe80::213:c4ff:fedd:14b0%vlan10 deleted
```

Figure 2-3. Example of Clearing the IPv6 Neighbors Cache

Telnet6 Operation

This section describes Telnet operation for IPv6 on the switch. For IPv4 Telnet operation, refer to the *Management and Configuration Guide* for your switch.

Outbound Telnet6 to Another Device

Syntax: telnet < *link-local-addr* >%vlan< *vid* >
telnet < *global-unicast-addr* >

Outbound Telnet6 establishes a Telnet session from the switch CLI to another IPv6 device, and includes these options.

- *Telnet for Link-Local Addresses on the same VLAN requires the link-local address and interface scope:*

  < *link-local-addr* >: *Specifies the link-local IPv6 address of the destination device.*

  %vlan< *vid* >: *Suffix specifying the interface on which the destination device is located. No spaces are allowed in the suffix.*

- *Telnet for Global Unicast Addresses requires a global unicast address for the destination. Also, the switch must be receiving router advertisements from an IPv6 gateway router.*

  < *global-unicast-addr* >: *Specifies the global IPv6 address of the destination device.*

For example, to Telnet to another IPv6 device having a link-local address of fe80::215:60ff:fe79:8980 and on the same VLAN interface (VLAN 10), you would use the following command:

```
HP Switch(config)# telnet fe80::215:60ff:fe79:980%vlan10
```

If the switch is receiving router advertisements from an IPv6 default gateway router, you can Telnet to a device on the same VLAN or another VLAN or subnet by using its global unicast address. For example, to Telnet to a device having an IPv6 global unicast address of 2001:db8::215:60ff:fe79:980, you would enter the following command:

```
HP Switch(config)# telnet 2001:db8::215:60ff:fe79:980
```

Viewing the Current Telnet Activity on a Switch

Syntax: show telnet

This command shows the active incoming and outgoing telnet sessions on the switch (for both IPv4 and IPv6). Command output includes the following:

Session: *The session number. The switch allows one outbound session and up to five inbound sessions.*

Privilege: *Manager or Operator.*

From: *Console (for outbound sessions) or the source IP address of the inbound session.*

To: *The destination of the outbound session, if in use.*

For example, the following figure shows that the switch is running one outbound, IPv4 session and is being accessed by two inbound sessions.

```
HP Switch# show telnet

 Telnet Activity

 ------------------------------------------------------
 Session  :     1
 Privilege: Manager
 From     : Console
 To       : 10.0.10.140
 ------------------------------------------------------
 Session  :     2
 Privilege: Manager
 From     : 2620:0:260:212::2:219
 To       :
 ------------------------------------------------------
 Session  : **  3
 Privilege: Manager
 From     : fe80::2:101
 To       :
```

The ** in the "Session: indicates the session through which **show telnet** was run.

Figure 2-4. Example of Show Telnet Output with Three Sessions Active

Enabling or Disabling Inbound Telnet Access

Syntax: [no] telnet-server

This command is used at the global config level to enable (the default) or disable all (IPv4 and IPv6) inbound Telnet access to the switch.

The **no** *form of the command disables inbound telnet.*

Note: *To disable inbound Telnet access completely, you must disable Telnet access for both IPv6 and IPv4. (The command for disabling Telnet4 access is* **no telnet-server**.*)*

For example, to disable Telnet6 access to the switch, you would use this command:

```
HP Switch(config)# no telnet-server
```

Viewing the Current Inbound Telnet Configuration

Syntax: show console

This command shows the current configuration of IPv4 and IPv6 inbound telnet permissions, as well as other information. For both protocols, the default setting allows inbound sessions.

```
HP Switch(config)# show console

 Console/Serial Link

  USB Console Input Enabled [Yes] : Yes
  Inbound Telnet Enabled [Yes] : Yes
  Web Agent Enabled [Yes] : Yes

  Terminal Type [VT100] : VT100
  Screen Refresh Interval (sec) [3] : 3
  Displayed Events [All] : All
  Baud Rate [Speed Sense] : speed-sense
  Flow Control [XON/XOFF] : XON/XOFF
  Session Inactivity Time (min) [0] : 0
```

Inbound Telnet Setting for IPv4 and IPv6 Telnet

Figure 2-5. Show Console Output Showing Default Console Configuration

SNTP and Timep

Configuring (Enabling or Disabling) the SNTP Mode

This section lists the SNTP and related commands, including an example of using an IPv6 address. For the details of configuring SNTP on the switch, refer to the chapter titled "Time Protocols" in the *Management and Configuration Guide* for your switch.

The following commands are available at the global config level for SNTP operation.

Commands Affecting SNTP	Function
show sntp	Display the current SNTP configuration.
timesync < sntp | timep >	Enable either SNTP or Timep as the time synchronization method on the switch without affecting the configuration of either.
[no] timesync	Enable time synchronization. (Requires a timesync method to also be enabled.) The no version disable time synchronization without affecting the configuration of the current time synchronization method.)
[no]sntp	Enables SNTP with the current SNTP configuration. The no version disables SNTP without changing the current SNTP configuration.
sntp < unicast | broadcast >	Configures the SNTP mode. (Default: Broadcast)
sntp < 30 - 720 >	Changes the interval between time requests. (Default: 720 seconds)

Configuring an IPv6 Address for an SNTP Server

Note

To use a global unicast IPv6 address to configure an IPv6 SNTP time server on the switch, the switch must be receiving advertisements from an IPv6 router on a VLAN configured on the switch.

To use a link-local IPv6 address to configure an IPv6 SNTP time server on the switch, it is necessary to append **%vlan** followed immediately (without spaces) by the VLAN ID of the VLAN on which the server address is available. (The VLAN must be configured on the switch.) For example:

```
fe80::11:215%vlan10
```

Syntax:. [no] sntp server priority < 1 - 3 > < *link-local-addr* >%vlan< *vid* > [1 - 7]
[no] sntp server priority < 1 - 3 > < *global-unicast-addr* > [1 - 7]

Configures an IPv6 address for an SNTP server.

server priority < 1 - 3 >: *Specifies the priority of the server addressing being configured. When the SNTP mode is set to unicast and more than one server is configured, this value determines the order in which the configured servers will be accessed for a time value. The switch polls multiple servers in order until a response is received or all servers on the list have been tried without success. Up to three server addresses (IPv6 and/or IPv4) can be configured.*

< link-local-addr >: *Specifies the link-local IPv6 address of the destination device.*

%vlan< *vid* >: *Suffix specifying the interface on which the destination device is located. No spaces are allowed in the suffix.*

< global-unicast-addr >: *Specifies the global IPv6 address of the destination device.*

[1 - 7]: *This optional setting specifies the SNTP server version expected for the specified server. (Default: 3)*

For example, to configure link-local and global unicast SNTP server addresses of:

- fe80::215:60ff:fe7a:adc0 (on VLAN 10, configured on the switch)
- 2001:db8::215:60ff:fe79:8980

as the priority "1" and "2" SNTP servers, respectively, using version 7, you would enter these commands at the global config level, as shown below.

```
HP Switch(config)# sntp server priority 1
fe80::215:60ff:fe7a:adc0%vlan10 7

HP Switch(config)# sntp server priority 2
2001:db8::215:60ff:fe79:8980 7
```

Note

In the preceding example, using a link-local address requires that you specify the local scope for the address; VLAN 10 in this case. This is always indicated by **%vlan** followed immediately (without spaces) by the VLAN identifier.

Syntax:. show sntp

Displays the current SNTP configuration, including the following:

Time Sync Mode: *Indicates whether timesync is disabled or set to either SNTP or Timep. (Default:* **timep***)*

SNTP Mode: *Indicates whether SNTP uses the broadcast or unicast method of contacting a time server. The broadcast option does not require you to configure a time server address. The unicast option does require configuration of a time server address.*

Poll Interval: *Indicates the interval between consecutive time requests to an SNTP server.*

Priority: *Indicates the configured priority for the corresponding SNTP server address.*

SNTP Server Address: *Lists the currently configured SNTP server addresses.*

Protocol Version: *Lists the SNTP server protocol version to expect from the server at the corresponding address.*

For example, the **show sntp** output for the preceding **sntp server** command example would appear as follows:

```
HP Switch(config)# show sntp

 SNTP Configuration

  Time Sync Mode: Sntp
  SNTP Mode : Broadcast
  Poll Interval (sec) [720] : 719

  Priority SNTP Server Address                          Protocol Version
  -------- ------------------------------------------- ----------------
  1        2001:db8::215:60ff:fe79:8980                 7
  2        10.255.5.24                                  3
```

This example illustrates the command output when both IPv6 and IPv4 server addresses are configured.

Figure 2-6. Example of Show SNTP Output with Both an IPv6 and an IPv4 Server Address Configured

Note that the **show management** command can also be used to display SNTP server information.

Configuring (Enabling or Disabling) the Timep Mode

This section lists the Timep and related commands, including an example of using an IPv6 address. For the details of configuring Timep on the switch, refer to the chapter titled "Time Protocols" in the *Management and Configuration Guide* for your switch.

The following commands are available at the global config level for Timep operation.

Commands Affecting Timep	Function
show timep	Display the current timep configuration.
timesync < sntp | timep >	Enable either SNTP or Timep as the time synchronization method on the switch without affecting the configuration of either.
ip timep dhcp [interval < 1 - 9999 >]	Enable Timep operation with a Timep server assignment configured from an IPv4 or IPv6 DHCP server. Optionally change the interval between time requests.
ip timep manual < *ipv6-addr* > [interval < 1 - 9999 >]	Enable Timep operation with a statically configured IPv6 address for a Timep server. Optionally change the interval between time requests.

no ip timep	Disables Timep operation. To re-enable Timep, it is necessary to reconfigure either the DHCP or the static option.

Note

To use a global unicast IPv6 address to configure an IPv6 Timep server on the switch, the switch must be receiving advertisements from an IPv6 router on a VLAN configured on the switch.

To use a link-local IPv6 address to configure an IPv6 Timep server on the switch, it is necessary to append **%vlan** followed (without spaces) by the VLAN ID of the VLAN on which the server address is available. The VLAN must be configured on the switch. For example: `fe80::11:215%vlan10`

Syntax:. ip timep dhcp [interval < 1 - 9999 >]
ip timep manual < *ipv6-addr* | *ipv4-addr* > [interval < 1 - 9999 >]

Used at the global config level to configure a Timep server address.

Note: *The switch allows one Timep server configuration.*

timep dhcp: *Configures the switch to obtain the address of a Timep server from an IPv4 or IPv6 DHCP server.*

timep manual: *Specifies static configuration of a Timep server address.*

< *ipv6-addr* >: *Specifies the IPv6 address of an SNTP server. Refer to preceding* **Note**.

[Interval < 1 - 9999 >]: *This optional setting specifies the interval in minutes between Timep requests. (Default: 720)*

For example, to configure a link-local Timep server address of:

fe80::215:60ff:fe7a:adc0

where the address is on VLAN 10, configured on the switch, you would enter this command at the global config level, as shown below.

```
HP Switch(config)# ip timep manual
fe80::215:60ff:fe7a:adc0%vlan10
```

Note

In the preceding example, using a link-local address requires that you specify the local scope for the address; VLAN 10 in this case. This is always indicated by **%vlan** followed immediately (without spaces) by the VLAN identifier. For a global unicast address, you would enter the address *without* the **%vlan** suffix.

Syntax:. show timep

Displays the current Timep configuration, including the following:

Time Sync Mode: *Indicates whether timesync is disabled or set to either SNTP or Timep. (Default: Disabled)*

Timep Mode: *Indicates whether Timep is configured to use a DHCP server to acquire a Timep server address or to use a statically configured Timep server address.*

Server Address: *Lists the currently configured Timep server address.*

Poll Interval (min) [720]: *Indicates the interval between consecutive time requests to the configured Timep server.*

For example, the **show timep** output for the preceding **ip timep manual** command example would appear as follows:

```
HP Switch(config)# sho timep

 Timep Configuration

  Time Sync Mode: Timep
  TimeP Mode [Disabled] : Manual
  Server Address : fe80::215:60ff:fe7a:adc0%vlan10
  Poll Interval (min) [720] : 720
```

Figure 2-7. Example of Show Timep Output with an IPv6 Server Address Configured

Note that the **show management** command can also be used to display Timep server information.

TFTP File Transfers Over IPv6

TFTP File Transfers over IPv6

You can use TFTP **copy** commands over IPv6 to upload, or download files to and from a physically connected device or a remote TFTP server, including:

- Switch software
- Software images
- Switch configurations
- ACL command files
- Diagnostic data (crash data, crash log, and event log)

For complete information on how to configure TFTP file transfers between the switch and a TFTP server or other host device on the network, refer to the "File Transfers" appendix in the *Management and Configuration Guide* for your switch.

To upload and/or download files to the switch using TFTP in an IPv6 network, you must:

1. Enable TFTP for IPv6 on the switch (see "Enabling TFTP for IPv6" on page 2-15).
2. Enter a TFTP **copy** command with the IPv6 address of a TFTP server in the command syntax (see "Using TFTP to Copy Files over IPv6" on page 2-16).
3. (Optional) To enable auto-TFTP operation, enter the **auto-tftp** command (see "Using Auto-TFTP for IPv6" on page 2-19).

Enabling TFTP for IPv6

Client and server TFTP for IPv6 is enabled by default on the switch. However, if it is disabled, you can re-enable it by specifying TFTP client or server functionality with the **tftp** <**client** | **server**> command. Enter the **tftp** < **client** | **server**> command at the global configuration level.

Syntax: [no] tftp <client | server >

Enables TFTP for IPv4 and IPv6 client or server functionality so that the switch can:

- *Use TFTP client functionality to access IPv4- or IPv6-based TFTP servers in the network to receive downloaded files.*
- *Use TFTP server functionality on the switch to be accessed by other IPv4 or IPv6 hosts requesting to upload files.*

The **no** *form of the command disables the client or server functionality.*

Default: TFTP Client and Server functionality enabled

Usage Notes

To disable all TFTP client or server operation on the switch except for the auto-TFTP feature, enter the **no tftp** <**client** | **server**> command. To re-enable TFTP client or server operation, re-enter the **tftp** <**client** | **server**> command. (Entering **no tftp** without specifying client or server affects only the client functionality. To disable or re-enable the TFTP server functionality, you must specify **server** in the command.)

When TFTP is disabled, instances of TFTP in the CLI **copy** command and the Menu interface “Download OS” screen become unavailable.

The **[no] tftp** <**client** | **server**> command does not affect auto-TFTP operation. For more information, see “Using Auto-TFTP for IPv6” on page 2-19.

Using TFTP to Copy Files over IPv6

Use the TFTP **copy** commands described in this section to:

- Download specified files from a TFTP server to a switch on which TFTP client functionality is enabled.
- Upload specified files from a switch, on which TFTP server functionality is enabled, to a TFTP server.

Syntax: copy tftp < *target* > < *ipv6-addr* > < *filename* >

Copies (downloads) a data file from a TFTP server at the specified IPv6 address to a target file on a switch that is enabled with TFTP server functionality.

< *ipv6-addr* >: *If this is a link-local address, use this IPv6 address format:*

fe80::< *device-id* >%vlan< *vid* >

For example: `fe80::123%vlan10`

If this is a global unicast address, use this IPv6 format:

< *ipv6-addr* >

For example: `2001:db8::123`

< *target* > *is one of the following values:*

- **autorun-cert-file**: Copies an autorun trusted certificate to the switch.
- **autorun-key-file**: Copies an autorun key file to the switch.
- **command-file**: Copies a file stored on a remote host and executes the ACL command script on the switch. Depending on the ACL commands stored in the file, one of the following actions is performed in the running-config file on the switch:
  - *A new ACL is created.*
  - *An existing ACL is replaced.*
  - **match**, **permit**, *or* **deny** *statements are added to an existing ACL.*

  For more information on ACLs, refer to "Creating an ACL Offline" in the Access Control Lists (ACLs) chapter in the Access Security Guide.
- **config < *filename* >:** *Copies the contents of a file on a remote host to a configuration file on the switch.*

- **flash** < **primary** | **secondary** >: *Copies a software file stored on a remote host to primary or secondary flash memory on the switch. To run a newly downloaded software image, enter the* **reload** *or* **boot system flash** *command.*
- **pub-key-file**: *Copies a public-key file to the switch.*
- **startup-config**: *Copies a configuration file on a remote host to the startup configuration file on the switch.*

Syntax: copy tftp <*source* >< *ipv6-addr* > < *filename* > < pc | unix >

Copies (uploads) a source data file on a switch that is enabled with TFTP server functionality to a file on the TFTP server at the specified IPv6 address, where <source> is one of the following values:

- **command-output < cli-command >**: *Copies the output of a CLI command to the specified file on a remote host.*
- **config < *filename* >:** *Copies the specified configuration file to a remote file on a TFTP server.*
- **crash-data < slot-id | master >:** *Copies the contents of the crash data file to the specified file path on a remote host. The crash data is software-specific and used to determine the cause of a system crash. You can copy crash information from an individual slot or from the master crash file on the switch.*
- **crash-log < *slot-id* | master >:** *Copies the contents of the crash log to the specified file path on a remote host. The crash log contains processor-specific operational data that is used to determine the cause of a system crash. You can copy the contents of the crash log from an individual slot or from the master crash log on the switch.*

- **event-log**: *Copies the contents of the Event Log on the switch to the specified file path on a remote host.*
- **flash < primary | secondary >:** *Copies the software file used as the primary or secondary flash image on the switch to a file on a remote host.*
- **startup-config**: *Copies the startup configuration file in flash memory to a remote file on a TFTP server.*
- **running-config**: *Copies the running configuration file to a remote file on a TFTP server.*

< *ipv6-addr* >: *If this is a link-local address, use this IPv6 address format:*

fe80::< *device-id* >%vlan< *vid* >

For example: `fe80::123%vlan10`

If this is a global unicast address, use this IPv6 format:

< *ipv6-addr* >

For example: `2001:db8::123`

Using Auto-TFTP for IPv6

At switch startup, the auto-TFTP for IPv6 feature automatically downloads a software image to the switch from a specified TFTP server, then reboots the switch. To implement the process the switch must first reboot using one of the following methods:

- enter the **boot system flash primary** command in the CLI
- with the default flash boot image set to primary flash (the default), enter the **boot** or the **reload** command, or cycle the power to the switch. (To reset the boot image to primary flash, use **boot set-default flash primary**.)

Syntax: auto-tftp <*ipv6-addr*> <*filename*>

Configures the switch to automatically download the specified software file from the TFTP server at the specified IPv6 address. The file is downloaded into primary flash memory at switch startup. The switch then automatically reboots from primary flash.

Notes: *To enable auto-TFTP to copy a software image to primary flash memory, the version number of the downloaded software file (for example, K_14_01.swi) must be different from the version number currently in the primary flash image.*

The current TFTP client status (enabled or disabled) does not affect auto-TFTP operation. (Refer to "Enabling TFTP for IPv6" on page 2-15.)

Completion of the auto-TFTP process may require several minutes while the switch executes the TFTP transfer to primary flash, and then reboots again.

The **no** *form of the command disables auto-TFTP operation by deleting the* **auto-tftp** *entry from the startup configuration. The* **no auto-tftp** *command does not affect the current TFTP-enabled configuration on the switch. However, entering the* **ip ssh filetransfer** *command automatically disables both* **auto-tftp** *and* **tftp** *operation.*

SNMP Management for IPv6

As with SNMP for IPv4, you can manage a switch via SNMP from an IPv6-based network management station by using an application such as IMC. (For more on IMC, go to the HP Networking web site at **www.hp.com\networking**.)

SNMP Features Supported

The same SNMP for IPv4 features are supported over IPv6:

- access to a switch using SNMP version 1, version 2c, or version 3
- enhanced security with the configuration of SNMP communities and SNMPv3 user-specific authentication password and privacy (encryption) settings
- SNMP notifications, including:
  - SNMP version 1 or SNMP version 2c traps
  - SNMPv2c informs
  - SNMPv3 notification process, including traps
- Advanced RMON (Remote Monitoring) management
- IMC management applications
- Flow sampling using sFlow
- Standard MIBs, such as the Bridge MIB (RFC 1493) and the Ethernet MAU MIB (RFC 1515)

SNMP Configuration Commands Supported

IPv6 addressing is supported in the following SNMP configuration commands: For more information on each SNMP configuration procedure, refer to the "Configuring for Network Management Applications" chapter in the current *Management and Configuration Guide* for your switch.

SNMPv1 and V2c

Syntax:. snmp-server host < *ipv4-addr* | *ipv6-addr* > < *community-name* > [none | all | non-info | critical | debug] [inform [retries < *count* >] [timeout < *interval* >]]

Executed at the global config level to configure an SNMP trap receiver to receive SNMPv1 and SNMPv2c traps, SNMPv2c informs, and (optionally) event log messages

SNMPv3

Syntax: snmpv3 targetaddress < *name* > params < *parms_name* > <*ipv4-addr* | *ipv6-addr*> [addr-mask < *ip4-addr* >] [filter < none | debug | all | not-info | critical>] [max-msg-size < 484-65535 >] [port-mask < *tcp-udp port* >] [retries < 0 - 255 >] [taglist <*tag_name*>] [timeout < 0 - 2147483647 >] [udp-port *port-number*]

Executed at the global config level to configure an SNMPv3 management station to which notifications (traps and informs) are sent.

Note

IPv6 is not supported in the configuration of an interface IPv6 address as the default source IP address used in the IP headers of SNMP notifications (traps and informs) or responses sent to SNMP requests. Only IPv4 addresses are supported in the following configuration commands:

snmp-server trap-source < *ipv4-addr* | loopback < 0-7 >>

snmp-server response-source [*dst-ip-of-request* | *ipv4-addr* | loopback < 0-7 >]

IPv6 addresses are supported in SNMP **show** command output as shown in Figure 2-8 and Figure 2-9.

The **show snmp-server** command displays the current SNMP policy configuration, including SNMP communities, network security notifications, link-change traps, trap receivers (including the IPv4 or IPv6 address) that can receive SNMPv1 and SNMPv2c traps, and the source IP (interface) address used in IP headers when sending SNMP notifications (traps and informs) or responses to SNMP requests.

```
HP Switch(config)# show snmp-server

 SNMP Communities

  Community Name       MIB View Write Access
  -------------------- -------- ------------
  public               Manager  Unrestricted
  marker               Manager  Unrestricted

 Trap Receivers

  Link-Change Traps Enabled on Ports [All] : All

  Traps Category                   Current Status
  ---------------------------      ---------------
  SNMP Authentication            : Extended
  Password change                : Enabled
  Login failures                 : Enabled
  Port-Security                  : Enabled
  Authorization Server Contact   : Enabled
  DHCP-Snooping                  : Enabled
  Dynamic ARP Protection         : Enabled

  Address                 Community              Events   Type   Retry   Timeout
  ----------------------- ---------------------- -------- ------ ------- -------
  15.29.17.218            public                 All      trap   3       15
  15.29.17.219            public                 Critical trap   3       15
  2620:0000:0260:0211
    :0217:a4ff:feff:1f70  marker                 Critical trap   3       15

 Excluded MIBs


 Snmp Response Pdu Source-IP Information

  Selection Policy   : rfc1517

 Trap Pdu Source-IP Information

  Selection Policy   : rfc1517
```

An IPv6 address is displayed on two lines.

Figure 2-8. "show snmp-server" Command Output with IPv6 Address

The **show snmpv3 targetaddress** command displays the configuration (including the IPv4 or IPv6 address) of the SNMPv3 management stations to which notification messages are sent.

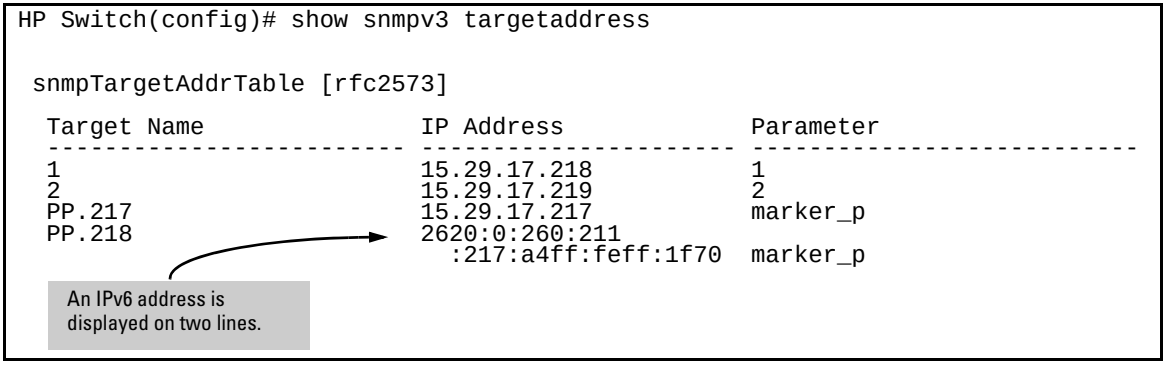


Figure 2-9. "show snmpv3 targetaddress" Command Output with IPv6 Address

IP Preserve for IPv6

IPv6 supports the IP Preserve feature, which allows you to copy a configuration file from a TFTP server to multiple switches without overwriting the IPv6 address and subnet mask on VLAN 1 (default VLAN) in each switch, and the Gateway IPv6 address assigned to the switch.

To configure IP Preserve, enter the **ip preserve** statement at the end of the configuration file that will be downloaded from a TFTP server. (Note that you do not invoke IP Preserve by entering a command from the CLI).

```
; J8697A Configuration Editor; Created on release #YA.15.xx
hostname "HPSwitch"
time daylight-time-rule None

   *
   *
   *
   *
   *
   *
password manager
password operator
ip preserve
```

Entering an **ip preserve** statement as the last line in a configuration file stored on a TFTP server allows you to download and execute the file as the startup-config file on an IPv6 switch. When the switch reboots, the configuration settings in the downloaded file are implemented without changing the IPv6 address and gateway assigned to the switch as shown in Figure 2-11.

Figure 2-10. Example of How to Enter IP Preserve in a Configuration File

To download an IP Preserve configuration file to an IPv6-based switch, enter the TFTP **copy** command as described in "TFTP File Transfers over IPv6" on page 2-14 to copy the file as the new startup-config file on a switch.

When you download an IP Preserve configuration file, the following rules apply:

- If the switch's current IPv6 address for VLAN 1 was statically configured and not dynamically assigned by a DHCP/Bootp server, the switch reboots and retains its current IPv6 address, subnet mask, and gateway address. All other configuration settings in the downloaded configuration file are applied.
- If the switch's current IPv6 address for VLAN 1 was assigned from a DHCP server and not statically configured, IP Preserve is suspended. The IPv6 addressing specified in the downloaded configuration file is implemented when the switch copies the file and reboots.
  - If the downloaded file specifies DHCP/Bootp as the source for the IPv6 address of VLAN 1, the switch uses the IPv6 address assigned by the DHCP/Bootp server.
  - If the file specifies a dedicated IPv6 address and subnet mask for VLAN 1 and a Gateway IPv6 address, the switch implements these settings in the startup-config file.

To verify how IP Preserve was implemented in a switch, after the switch reboots, enter the **show run** command. Figure 2-11 shows an example in which all configurations settings have been copied into the startup-config file except for the IPv6 address of VLAN 1 (2001:db8::214:c2ff:fe4c:e480) and the default IPv6 gateway (2001:db8:0:7::5), which were retained.

Note that if a switch received its IPv6 address from a DHCP server, the "ip address" field under "vlan 1" would display: **dhcp-bootp**.

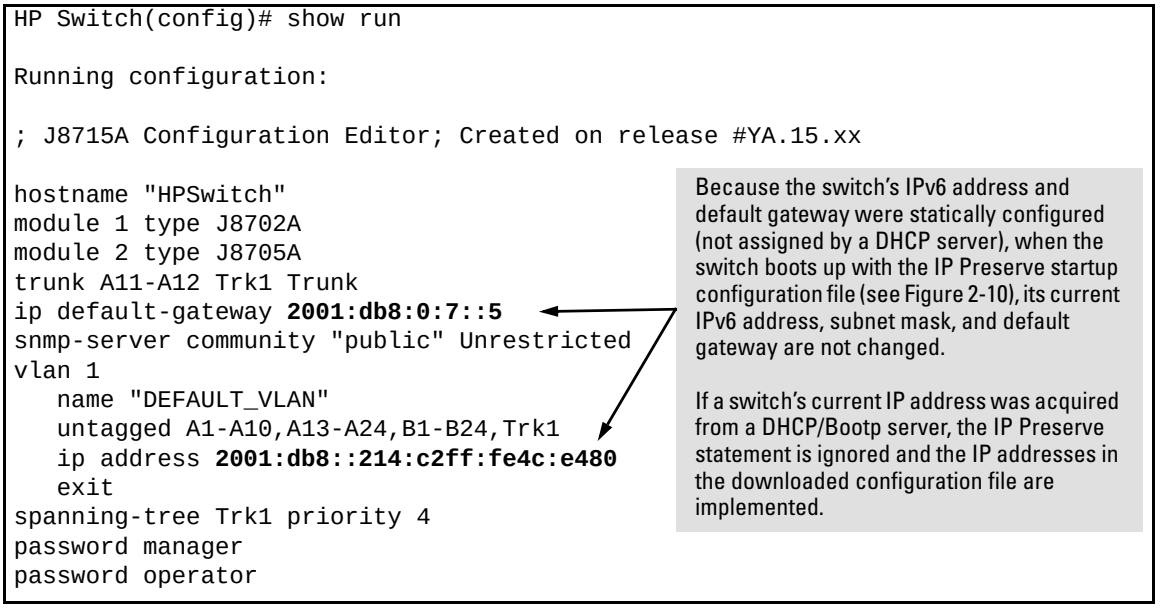


Figure 2-11. Configuration File with Dedicated IP Addressing After Startup with IP Preserve

For more information on how to use the IP Preserve feature, refer to the "Configuring IP Addressing" chapter in the current *Management and Configuration Guide* for your HP switch.

IPv6 Management Security Features

IPv6 Management Security

This chapter describes management security features that are IPv6 counterparts of IPv4 management security features on the switches covered by this guide.

Feature	Default	CLI
configure authorized IP managers for IPv6	disabled	3-4
enabling secure copy and secure FTP for IPv6	disabled	3-14

This chapter describes the following IPv6-enabled management security features:

- Authorized IP Managers for IPv6
- Secure Copy and Secure FTP for IPv6

Authorized IP Managers for IPv6

The Authorized IP Managers feature uses IP addresses and masks to determine which stations (PCs or workstations) can access the switch through the network. This feature supports switch access through:

- Telnet and other terminal emulation applications
- Web browser interface
- SNMP (with a correct community name)

As with the configuration of IPv4 management stations, the Authorized IP Managers for IPv6 feature allows you to specify the IPv6-based stations that can access the switch.

Usage Notes

- You can configure up to ten authorized IPv4 and IPv6 manager *addresses* on a switch, where each address applies to either a single management station or a group of stations. Each authorized manager address consists of an IPv4 or IPv6 address and a mask that determines the individual management stations that are allowed access.
  - You configure authorized IPv4 manager addresses using the **ip authorized-managers** command. For more information, refer to the "Using Authorized IP Managers" chapter in the *Access Security Guide*.
  - You configure authorized IPv6 manager addresses using the **ipv6 authorized-managers** command. For more information, see "Configuring Authorized IP Managers for Switch Access" on page 3-4.
- You can block all IPv4-based or all IPv6-based management stations from accessing the switch by entering the following commands:
  - To block access to all IPv4 manager addresses while allowing access to IPv6 manager addresses, enter the **ip authorized-managers 0.0.0.0** command.
  - To block access to all IPv6 manager addresses while allowing access to IPv4 manager addresses, enter the **ipv6 authorized-managers ::** command. (The double colon represents an IPv6 address that consists of all zero's: **0:0:0:0:0:0:0:0**.)

- You configure each authorized manager address with Manager or Operator-level privilege to access the switch in a Telnet, SNMPv1, or SNMPv2c session. (Access privilege for SSH, SNMPv3, and web browser sessions are configured through the access application, not through the Authorized IP Managers feature.)
  - Manager privilege allows full access to all web browser and console interface screens for viewing, configuration, and all other operations available in these interfaces.
  - Operator privilege allows read-only access from the web browser and console interfaces.
- When you configure station access to the switch using the Authorized IP Managers feature, the settings take precedence over the access configured with local passwords, TACACS+ servers, RADIUS-assigned settings, port-based (802.1X) authentication, and port security settings.

  As a result, the IPv6 address of a networked management device must be configured with the Authorized IP Managers feature before the switch can authenticate the device using the configured settings from other access security features. If the Authorized IP Managers feature disallows access to the device, then access is denied. Therefore, with authorized IP managers configured, logging in with the correct passwords is not sufficient to access a switch through the network unless the station requesting access is also authorized in the switch's Authorized IP Managers configuration.

Configuring Authorized IP Managers for Switch Access

To configure one or more IPv6-based management stations to access the switch using the Authorized IP Managers feature, enter the **ipv6 authorized-managers** command

Syntax: ipv6 authorized-managers *<ipv6-addr>* [*ipv6-mask*] [access <operator | manager>]

Configures one or more authorized IPv6 addresses to access the switch, where:

ipv6-mask *specifies the mask that is applied to an IPv6 address to determine authorized stations. For more information, see "Using a Mask to Configure Authorized Management Stations" on page 3-4. Default:* **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF**.

access <operator | manager> *specifies the level of access privilege granted to authorized stations and applies only to Telnet, SNMPv1, and SNMPv2c access. Default:* **Manager.**

Note: *The Authorized IP Manager feature does not support the configuration of access privileges on authorized stations that use an SSH, SNMPv3, or the web browser session to access the switch. For these sessions, access privilege is configured with the access application.*

Using a Mask to Configure Authorized Management Stations

The *ipv6-mask* parameter controls how the switch uses an IPv6 address to determine the IPv6 addresses of authorized manager stations on your network. For example, you can specify a mask that authorizes:

- Single station access
- Multiple station access

Note

Mask configuration is a method for determining the valid IPv6 addresses that are authorized for management access to the switch. In the Authorized IP Managers feature, the mask serves a different purpose than an IPv6 subnet mask and is applied in a different manner.

Configuring Single Station Access

To authorize only one IPv6-based station for access to the switch, enter the IPv6 address of the station and set the mask to **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF**.

Notes

If you do not enter a value for the *ipv6-mask* parameter when you configure an authorized IPv6 address, the switch automatically uses **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF** as the default mask (see "Configuring Authorized IP Managers for Switch Access" on page 3-4).

If you have ten or fewer management and/or operator stations for which you want to authorize access to the switch, it may be more efficient to configure them by entering each IPv6 address with the default mask in a separate **ipv6 authorized-managers** command.

When used in a mask, "**FFFF**" specifies that each bit in the corresponding 16-bit (hexadecimal) block of an authorized station's IPv6 address must be identical to the same "on" or "off" setting in the IPv6 address entered in the **ipv6 authorized-managers** command. (The binary equivalent of **FFFF** is 1111 1111 1111 1111, where **1** requires the same "on" or "off" setting in an authorized address.)

For example, as shown in Figure 3-1, if you configure a link-local IPv6 address of FE80::202:B3FF:FE1E:8329 with a mask of **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF**, only a station having an IPv6 address of FE80::202:B3FF:FE1E:8329 has management access to the switch.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block	Manager- or Operator-Level Access
IPv6 Mask	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	The "FFFF" in each hexadecimal block of the mask specifies that only the exact value of each bit in the corresponding block of the IPv6 address is allowed. This mask allows management access only to a station having an IPv6 address of FE80::202:B3FF:FE1E:8329.
IPv6 Address	FE80	0000	0000	0000	202	B3FF	FE1E	8329	

Figure 3-1. Mask for Configuring a Single Authorized IPv6 Manager Station

Configuring Multiple Station Access

To authorize multiple stations to access the switch without having to re-enter the **ipv6 authorized-managers** command for each station, carefully select the IPv6 address of an authorized IPv6 manager and an associated mask to authorize a range of IPv6 addresses.

As shown in Figure 3-2, if a bit in any of the 4-bit binary representations of a hexadecimal value in a mask is "on" (set to 1), then the corresponding bit in the IPv6 address of an authorized station must match the "on" or "off' setting of the same bit in the IPv6 address you enter with the **ipv6 authorized-managers** command.

Conversely, in a mask, a "0" binary bit means that either the "on" or "off" setting of the corresponding IPv6 bit in an authorized address is valid and does not have to match the setting of the same bit in the specified IPv6 address.

Figure 3-2 shows the binary expressions represented by individual hexadecimal values in an *ipv6-mask* parameter.

Hexadecimal Value in an IPv6 Mask	Binary Equivalent
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Figure 3-2. Hexadecimal Mask Values and Binary Equivalents

Example. Figure 3-3 shows an example in which a mask that authorizes switch access to four management stations is applied to the IPv6 address: **2001:DB8:0000:0000:244:17FF:FEB6:D37D**. The mask is: **FFFF:FFFF:FFFF:FFF8:FFFF:FFFF:FFFF:FFFC**.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block	Manager- or Operator-Level Access
IPv6 Mask	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFC	The "F" value in the first 124 bits of the mask specifies that only the exact value of each corresponding bit in an authorized IPv6 address is allowed. However, the "C" value in the last four bits of the mask allows four possible combinations (D37C, D37D, D37E, and D37F) in the last block of an authorized IPv6 address.
IPv6 Address	2001	DB8	0000	0000	244	17FF	FEB6	D37D	

Figure 3-3. Example: Mask for Configuring Four Authorized IPv6 Manager Stations

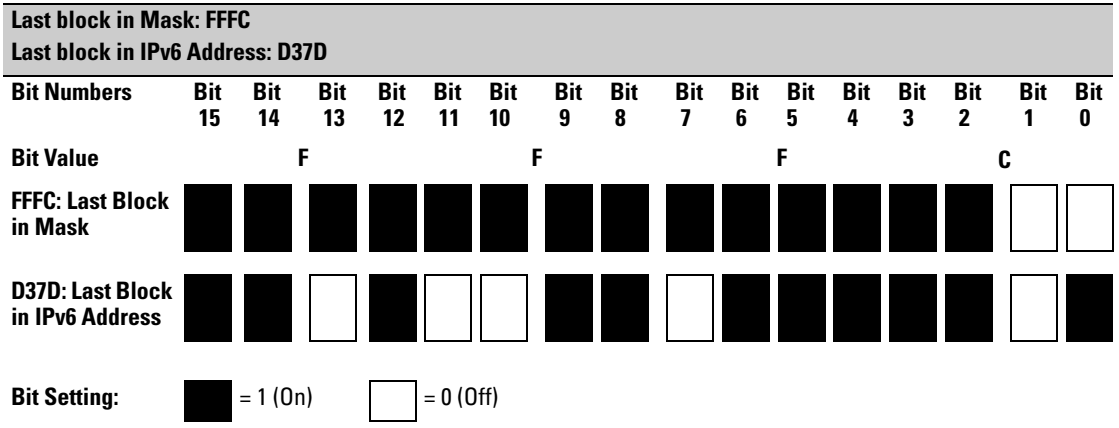


Figure 3-4. Example: How a Mask Determines Four Authorized IPv6 Manager Addresses

As shown in Figure 3-4, if you use a mask of **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFC** with an IPv6 address, you can authorize four IPv6-based stations to access the switch. In this mask, all bits except the last two are set to 1 ("on"); the binary equivalent of hexadecimal **C** is 1100.

Therefore, this mask requires the first corresponding 126 bits in an authorized IPv6 address to be the same as in the specified IPv6 address: 2001:DB8:0000:0000:244:17FF:FEB6:D37C. However, the last two bits are set

to 0 (“off”) and allow the corresponding bits in an authorized IPv6 address to be either “on” or “off”. As a result, only the four IPv6 addresses shown in Figure 3-5 are allowed access.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block
IPv6 Mask	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFC
IPv6 Address Entered with the “ipv6 authorized-managers” Command	2001	DB8	0000	0000	244	17FF	FEB6	D37D
Other Authorized IPv6 Addresses	2001	DB8	0000	0000	244	17FF	FEB6	D37C
	2001	DB8	0000	0000	244	17FF	FEB6	D37E
	2001	DB8	0000	0000	244	17FF	FEB6	D37F

Figure 3-5. Example: How Hexadecimal C in a Mask Authorizes Four IPv6 Manager Addresses

Example. Figure 3-6 shows an example in which a mask is applied to the IPv6 address: **2001:DB8:0000:0000:244:17FF:FEB6:D37D/64**. The specified mask **FFFF:FFFF:FFFF:FFF8:FFFF:FFFF:FFFF:FFFF** configures eight management stations as authorized IP manager stations.

Note that, in this example, the IPv6 mask is applied as follows:

- Eight management stations in different subnets are authorized by the value of the fourth block (**FFF8**) in the 64-bit prefix ID (**FFFF:FFFF:FFFF:FFF8**) of the mask. (The fourth block of the prefix ID is often used to define subnets in an IPv6 network.)

  The binary equivalent of **FFF8** that is used to specify valid subnet IDs in the IPv6 addresses of authorized stations is: 1111 1111 1111 1000.

  The three “off” bits (1000) in the last part of the this block (**FFF8**) of the mask allow for eight possible authorized IPv6 stations:
  2001:DB8:0000:0000:244:17FF:FEB6:D37D
  2001:DB8:0000:0001:244:17FF:FEB6:D37D
  2001:DB8:0000:0002:244:17FF:FEB6:D37D
  2001:DB8:0000:0003:244:17FF:FEB6:D37D
  2001:DB8:0000:0004:244:17FF:FEB6:D37D
  2001:DB8:0000:0005:244:17FF:FEB6:D37D
  2001:DB8:0000:0006:244:17FF:FEB6:D37D
  2001:DB8:0000:0007:244:17FF:FEB6:D37D

- Each authorized station has the same 64-bit device ID (**244:17FF:FEB6:D37D**) because the value of the last four blocks in the mask is **FFFF** (binary value 1111 1111).

  FFFF requires all bits in each corresponding block of an authorized IPv6 address to have the same "on" or "off" setting as the device ID in the specified IPv6 address. In this case, each bit in the device ID (last four blocks) in an authorized IPv6 address is fixed and can be only one value: 244:17FF:FEB6:D37D.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block	Manager- or Operator-Level Access
IPv6 Mask	FFFF	FFFF	FFFF	FFF8	FFFF	FFFF	FFFF	FFFF	In this example, the IPv6 mask allows up to four stations in different subnets to access the switch. This authorized IP manager configuration is useful if only management stations are specified by the authorized IPv6 addresses. Refer to Figure 3-4 for how the bitmap of the IPv6 mask determines authorized IP manager stations.
Authorized IPv6 Address	2001	DB8	0000	0000	244	17FF	FEB6	D37D	

Figure 3-6. Example: Mask for Configuring Authorized IPv6 Manager Stations in Different Subnets

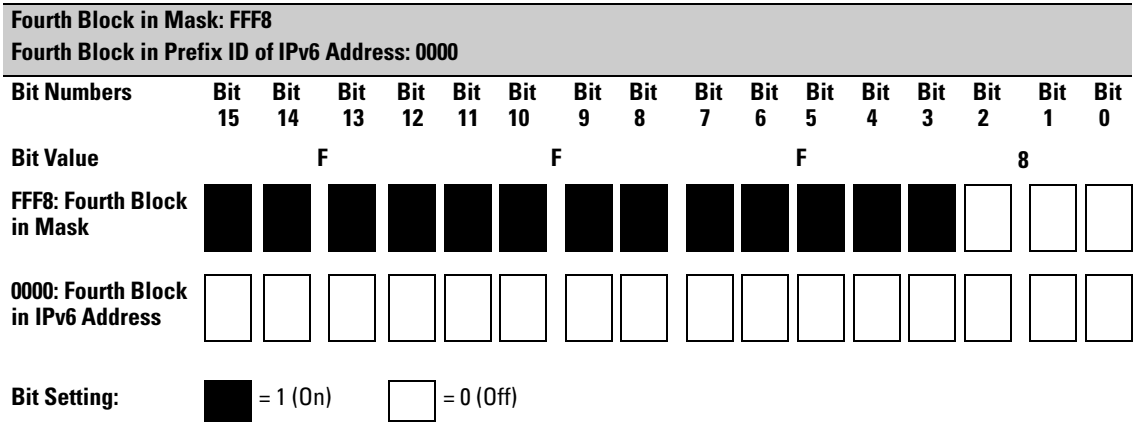


Figure 3-7. Example: How a Mask Determines Authorized IPv6 Manager Addresses by Subnet

Figure 3-7 shows the bits in the fourth block of the mask that determine the valid subnets in which authorized stations with an IPv6 device ID of **244:17FF:FEB6:D37D** reside.

FFF8 in the fourth block of the mask means that bits 3 - 15 of the block are fixed and, in an authorized IPv6 address, must correspond to the "on" and "off" settings shown for the binary equivalent 0000 in the fourth block of the IPv6 address. Conversely, bits 0 - 2 are variable and, in an authorized IPv6 address, may be either "on" (1) or "off" (0).

As a result, assuming that the seventh and eighth bytes (fourth hexadecimal block) of an IPv6 address are used as the subnet ID, only the following binary expressions and hexadecimal subnet IDs are supported in this authorized IPv6 manager configuration:

Authorized Subnet ID in Fourth Hexadecimal Block of IPv6 Address	Binary Equivalent
0000	0000 0000
0001	0000 0001
0002	0000 0010
0003	0000 0011
0004	0000 0100
0005	0000 0101
0006	0000 0110
0007	0000 0111

Figure 3-8. Binary Equivalents of Authorized Subnet IDs (in Hexadecimal)

Displaying an Authorized IP Managers Configuration

Use the **show ipv6 authorized-managers** command to list the IPv6 stations authorized to access the switch; for example:

```
HP Switch# show ipv6 authorized-managers

 IPv6 Authorized Managers
-------------------------------------

 Address : 2001:db8:0:7::5
 Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
 Access  : Manager

 Address : 2001:db8::a:1c:e3:3
 Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:fffe
 Access  : Manager

 Address : 2001:db8::214:c2ff:fe4c:e480
 Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
 Access  : Manager

 Address : 2001:db8::10
 Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff00
 Access  : Operator
```

Figure 3-9. Example of "show ipv6 authorized-managers" Output

By analyzing the masks displayed in Figure 3-9, the following IPv6 stations are granted access:

Mask	Authorized IPv6 Addresses	Number of Authorized Addresses
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFC	2001:db8:0:7::4 through 2001:db8:0:7::7	4
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFE	2001:db8::a:1c:e3:2 and 2001:db8::a:1c:e3:3	2
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF	2001:db8::214:c2ff:fe4c:e480	1
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FF00	2001:db8::0 through 2001:db8::FF	256

Figure 3-10. How Masks Determine Authorized IPv6 Manager Addresses

Additional Examples of Authorized IPv6 Managers Configuration

Authorizing Manager Access. The following IPv6 commands authorize manager-level access for one link-local station at a time. Note that when you enter a link-local IPv6 address with the **ipv6 authorized-managers** command, you must also enter a VLAN ID in the format: **%vlan**<*vlan-id*>.

```
ProCurve(config)# ipv6 authorized-managers
fe80::07be:44ff:fec5:c965%vlan2

ProCurve(config)# ipv6 authorized-managers
fe80::070a:294ff:fea4:733d%vlan2

ProCurve(config)# ipv6 authorized-managers
fe80::19af:2cff:fe34:b04a%vlan5
```

If you do not enter an *ipv6-mask* value when you configure an authorized IPv6 address, the switch automatically uses **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF** as the default IPv6 mask. Also, if you do not specify an **access** value to grant either Manager- or Operator-level access, by default, the switch assigns Manager access. For example:

```
HP Switch# ipv6 authorized-managers 2001:db8::a8:1c:e3:69
HP Switch# show ipv6 authorized-managers

 IPv6 Authorized Managers
--------------------------

Address : 2001:db8::a8:1c:e3:69
Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
Access  : Manager
```

If you do not enter a value for *ipv6-mask* in the **ipv6 authorized-managers** command, the default mask of FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF: is applied. The default mask authorizes only the specified station (see "Configuring Single Station Access" on page 3-4).

Figure 3-11. Default IPv6 Mask

The next IPv6 command authorizes operator-level access for sixty-four IPv6 stations: thirty-two stations in the subnets defined by 0x0006 and 0x0007 in the fourth block of an authorized IPv6 address:

```
ProCurve(config)# ipv6 authorized-managers
2001:db8:0000:0007:231:17ff:fec5:c967
ffff:ffff:ffff:fffe:ffff:ffff:ffff:ffe0 access operator
```

The following **ipv6 authorized-managers** command authorizes a single, automatically generated (EUI-64) IPv6 address with manager-level access privilege:

```
ProCurve(config)# ipv6 authorized-managers
::223:04ff:fe03:4501 ::ffff:ffff:ffff:ffff
```

Editing an Existing Authorized IP Manager Entry. To change the mask or access level for an existing authorized IP manager entry, enter the IPv6 address with the new value(s). Any parameters not included in the command are reset to their default values.

The following command replaces the existing mask and access level for IPv6 address 2001:DB8::231:17FF:FEC5:C967 with **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FF00** and **operator**:

```
ProCurve(config)# ipv6 authorized-managers
2001:db8::231:17ff:fec5:c967
ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff00 access operator
```

The following command replaces the existing mask and access level for IPv6 address 2001:DB8::231:17FF:FEC5:3E61 with **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF** and **manager** (the default values). Note that it is not necessary to enter either of these parameters:

```
ProCurve(config)# ipv6 authorized-managers
2001:db8::a05b:17ff:fec5:3f61
```

Deleting an Authorized IP Manager Entry. Enter only the IPv6 address of the configured authorized IP manager station that you want to delete with the **no** form of the command; for example:

```
ProCurve(config)# no ipv6 authorized-managers
2001:db8::231:17ff:fec5:3e61
```

Secure Copy and Secure FTP for IPv6

You can take advantage of the Secure Copy (SCP) and Secure FTP (SFTP) client applications to provide a secure alternative to TFTP for transferring sensitive switch information, such as configuration files and login information, between the switch and an administrator workstation.

SCP and SFTP run over an encrypted SSH session allowing you to use a secure SSH tunnel to:

- Transfer files and update HP software images.
- Distribute new software images with automated scripts that make it easier to upgrade multiple switches simultaneously and securely.

By default, SSH is enabled for IPv4 and IPv6 connections on a switch. If you have not disabled SSH connections from IPv6 clients (by entering the **ip ssh ip-version 4** command), you can perform secure file transfers to and from IPv6 client devices by entering the **ip ssh filetransfer** command.

Syntax:. [no] ip ssh filetransfer

Enables SSH on the switch to connect to an SCP or SFTP client application to transfer files to and from the switch.

Use the **no ip ssh filetransfer** *command to disable the switch's ability to perform secure file transfers with an SCP or SFTP client, without disabling SSH on the switch.*

After an IPv6 client running SCP/SFTP successfully authenticates and opens an SSH session on the switch, you can copy files to and from the switch using secure, encrypted file transfers. Refer to the documentation that comes with an SCP or SFTP client application for information on the file transfer commands and software utilities to use.

Notes

The switch supports one SFTP session or one SCP session at a time.

All files on the switch have read-write permission. However, several SFTP commands, such as **create** or **remove**, are not supported and return an error message.

For complete information on how to configure SCP or SFTP in an SSH session to copy files to and from the switch, refer to the *"File Transfers"* appendix in the *Management and Configuration Guide* for your switch.

Multicast Listener Discovery (MLD) Snooping

Overview

Multicast addressing allows one-to-many or many-to-many communication among hosts on a network. Typical applications of multicast communication include audio and video streaming, desktop conferencing, collaborative computing, and similar applications.

Multicast Listener Discovery (MLD) is an IPv6 protocol used on a local link for multicast group management. MLD is enabled per VLAN, and is analogous to the IPv4 IGMP protocol.

MLD snooping is a subset of the MLD protocol that operates at the port level and conserves network bandwidth by reducing the flooding of multicast IPv6 packets.

This chapter describes concepts of MLD snooping and the CLI commands available for configuring it and for viewing its status.

Introduction to MLD Snooping

There are several roles that network devices may play in an IPv6 multicast environment:

- **MLD host**—a network node that uses MLD to "join" (subscribe to) one or more multicast groups
- **multicast router**—a router that routes multicast traffic between subnets
- **querier**—a switch or multicast router that identifies MLD hosts by sending out MLD queries, to which the MLD hosts respond

Curiously enough, a network node that acts as a *source* of IPv6 multicast traffic is only an indirect participant in MLD snooping—it just provides multicast traffic, and MLD doesn't interact with it. (Note, however, that in an application like desktop conferencing a network node may act as both a source and an MLD host; but MLD interacts with that node only in its role as an MLD host.)

A source node creates multicast traffic by sending packets to a multicast address. In IPv6, addresses with the first eight bits set (that is, "FF" as the first two characters of the address) are multicast addresses, and any node that listens to such an address will receive the traffic sent to that address. Application software running on the source and destination systems cooperates to determine what multicast address to use. (Note that this is a function of the application software, not of MLD.)

For example, if several employees engage in a desktop conference across the network, they all need application software on their computers. At the start of the conference, the software on all the computers determines a multicast address of, say, FF3E:30:2001:DB8::101 for the conference. Then any traffic sent to that address can be received by all computers listening on that address.

General operation. Multicast communication can take place without MLD, and by default MLD is disabled. In that case, if a switch receives a packet with a multicast destination address, it floods the packet to all ports in the same VLAN (except the port that it came in on). Any network nodes that are listening to that multicast address will see the packet; all other hosts ignore the packet.

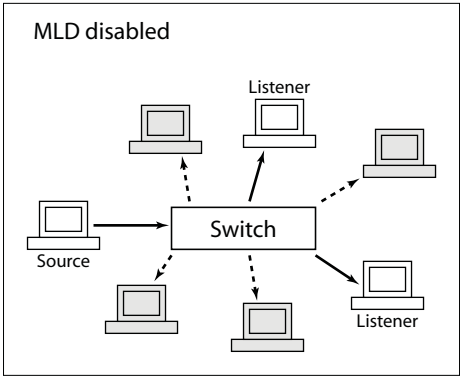


Figure 4-1. Without MLD, multicast traffic is flooded to all ports.

When MLD snooping is enabled on a VLAN, the switch acts to minimize unnecessary multicast traffic. If the switch receives multicast traffic destined for a given multicast address, it forwards that traffic only to ports on the VLAN that have MLD hosts for that address. It drops that traffic for ports on the VLAN that have no MLD hosts (except for a few special cases explained below).

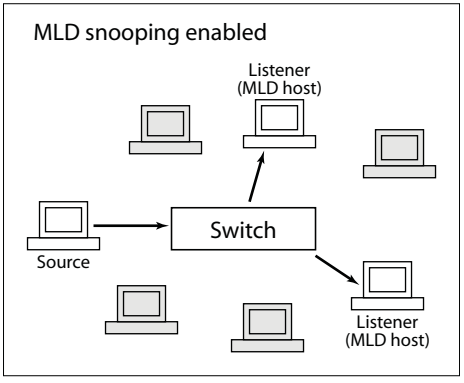


Figure 4-2. With MLD snooping, traffic is sent to MLD hosts.

Note that MLD snooping operates on a single VLAN (though there can be multiple VLANs, each running MLD snooping). Cross-VLAN traffic is handled by a multicast router.

Forwarding in MLD snooping. When MLD snooping is active, a multicast packet is handled by the switch as follows:

- forwarded to ports that have nodes that have joined the packet's multicast address (that is, MLD hosts on that address)
- forwarded toward the querier—If the switch is not the querier, the packet is forwarded out the port that leads to the querier.
- forwarded toward any multicast routers—If there are multicast routers on the VLAN, the packet is forwarded out any port that leads to a router.
- forwarded out administratively forwarded ports—The packet will be forwarded through all ports set administratively to forward mode. (See the description of forwarding modes, below.)
- dropped for all other ports

Each individual port's forwarding behavior can be explicitly set using a CLI command to one of these modes:

- auto (the default mode)—The switch forwards packets through this port based on the MLD rules and the packet's multicast address. In most cases, this means that the switch forwards the packet only if the port connects to a node that is joined to the packet's multicast address (that is, to an MLD host). There is seldom any reason to use a mode other than "auto" in normal operation (though some diagnostics may make use of "forward" or "block" mode).
- forward—The switch forwards all IPv6 multicast packets through the port. This includes IPv6 multicast data and MLD protocol packets.
- block—The switch drops all MLD packets received by the port and blocks all outgoing IPv6 multicast packets through the port, except those packets destined for well known IPv6 multicast addresses. This has the effect of preventing IPv6 multicast traffic from moving through the port.

Note that the switch floods all packets with "well known" IPv6 multicast destination addresses through all ports. Well known addresses are permanent addresses defined by the Internet Assigned Numbers Authority (www.iana.org). IPv6 standards define any address beginning with FF0x/12 (binary 1111 1111 0000) as a well known address.

Listeners and joins. The "snooping" part of MLD snooping arises because a switch must keep track of which ports have network nodes that are MLD hosts for any given multicast address. It does this by keeping track of "joins" on a per-port basis.

A network node establishes itself as an MLD host by issuing a multicast "join" request (also called a multicast "report") for a specific multicast address when it starts an application that listens to multicast traffic. The switch to which the node is connected sees the join request and forwards traffic for that multicast address to the node's port.

Queries. The querier is a multicast router or a switch that periodically asks MLD hosts on the network to verify their multicast join requests. There is one querier for each VLAN, and all switches on the VLAN listen to the responses of MLD hosts to multicast queries, and forward or block multicast traffic accordingly.

All of the HP Networking switches described by this guide have the querier function enabled by default. If there is another device on the VLAN that is already acting as querier, the switch defers to that querier. If there is no device acting as querier, the switch enters an election state and negotiates with other devices on the network (if any) to determine which one will act as the querier.

The querier periodically sends general queries to MLD hosts on each multicast address that is active on the VLAN. The time period that the querier waits between sending general queries is known as the query interval; the MLD standard sets the default query interval to 125 seconds.

Network nodes that wish to remain active as MLD hosts respond to the queries with join requests; in this way they continue to assert their presence as MLD hosts. The switch through which any given MLD host connects to the VLAN sees the join requests and continues forwarding traffic for that multicast address to the MLD host's port.

Leaves. A node acting as an MLD host can be disconnected from a multicast address in two ways:

- It can stop sending join requests to the querier. This might happen if the multicast application quits or the node is removed from the network. If the switch goes for slightly more than two query intervals without seeing a join request from the MLD host, it stops sending multicast traffic for that multicast address to the MLD host's port.
- It can issue a "leave" request. This is done by the application software running on the MLD host. If the MLD host is the only node connected to its switch port, the switch sees the leave request and stops sending multicast packets for that multicast address to that port. (If there is more than one node connected to the port the situation is somewhat more complicated, as explained below under "Fast leaves and forced fast leaves".)

Fast leaves and forced fast leaves. The fast leave and forced fast leave functions can help to prune unnecessary multicast traffic when an MLD host issues a leave request from a multicast address. Fast leave is enabled by default and forced fast leave is disabled by default. Both functions are applied to individual ports.

Which function to use depends on whether a port has more than one node attached to it, as follows:

- If a port has only one node attached to it, then when the switch sees a leave request from that node (an MLD host) it knows that it does not need to send any more multicast traffic for that multicast address to the host's port. If fast leave is enabled (the default setting), the switch stops sending the multicast traffic immediately. If fast leave is disabled, the switch continues to look for join requests from the host in response to group-specific queries sent to the port. The interval during which the switch looks for join requests is brief and depends on the forced fast leave setting: if forced fast leave is enabled for the port, it is equal to the "forced fast leave interval" (typically a couple of seconds or less); if forced fast leave is disabled for the port, the period is about 10 seconds (governed by the MLD standard). When this process has completed the multicast traffic for the group will be stopped (unless the switch sees a new join request).
- If there are multiple nodes attached to a single port, then a leave request from one of those nodes (an MLD host) does not provide enough information for the switch to stop sending multicast traffic to the port. In this situation the fast leave function does not operate. The switch continues to look for join requests from any MLD hosts connected to the port, in response to group-specific queries sent to the port. As in the case described above for a single-node port that is not enabled for fast leave, the interval during which the switch looks for join requests is brief and depends on the forced fast leave setting. If forced fast leave is enabled for the port, it is equal to the "forced fast leave interval" (typically a couple of seconds or less); if forced fast leave is disabled for the port, the period is about 10 seconds (governed by the MLD standard). When this process has completed the multicast traffic for the group will be stopped unless the switch sees a new join request. This reduces the number of multicast packets forwarded unnecessarily.

Configuring MLD

Several CLI commands are available for configuring MLD parameters on a switch.

Enabling or Disabling MLD Snooping on a VLAN

Syntax: [no] ipv6 mld <enable | disable>

Note: This command must be issued in a VLAN context.

This command enables MLD snooping on a VLAN. Enabling MLD snooping applies the last-saved or the default MLD configuration, whichever was most recently set.

The **[no]** *form of the command disables MLD snooping on a VLAN.*

MLD snooping is disabled by default.

For example, to enable MLD snooping on VLAN 8:

```
HP Switch# config
HP Switch(config)# vlan 8
(vlan-8)# ipv6 mld enable
```

To disable MLD snooping on VLAN 8:

```
HP Switch(vlan-8)# no ipv6 mld disable
```

Setting the MLD Version

Version 1 is the only version supported.

Syntax: [no] ipv6 mld version 1

Note: *This command must be issued in a VLAN context.*
Only version 1 is supported at this time.
Default: 1

Configuring Per-Port MLD Traffic Filters

Syntax: ipv6 mld [auto <*port-list*> | blocked <*port-list*> | forward <*port-list*>]

Note: *This command must be issued in a VLAN context.*

This command sets per-port traffic filters, which specify how each port should handle MLD traffic. Allowed settings are:

auto—*follows MLD snooping rules: packets are forwarded for joined groups*

blocked—*all multicast packets are dropped, except that packets for well known addresses are forwarded*

forward—*all multicast packets are forwarded*

The default value of the filter is **auto**.

<*port-list*>—*specifies the affected port or range of ports*

For example:

```
HP Switch(vlan-8)# ipv6 mld forward 16-18
HP Switch(vlan-8)# ipv6 mld blocked 19-21
HP Switch(vlan-8)# show ipv6 mld vlan 8 config

 MLD Service Vlan Config

  VLAN ID : 8
  VLAN NAME : VLAN8
  MLD Enabled [No] : Yes
  Querier Allowed [Yes] : Yes

  Port Type      | Port Mode Forced Fast Leave Fast Leave
  ---- --------- + --------- ----------------- ----------
  13   100/1000T | auto      No                Yes
  14   100/1000T | auto      No                Yes
  15   100/1000T | auto      No                Yes
  16   100/1000T | forward   No                Yes
  17   100/1000T | forward   No                Yes
  18   100/1000T | forward   No                Yes
  19   100/1000T | blocked   No                Yes
  20   100/1000T | blocked   No                Yes
  21   100/1000T | blocked   No                Yes
  22   100/1000T | auto      No                Yes
  23   100/1000T | auto      No                Yes
  24   100/1000T | auto      No                Yes
```

Figure 4-3. Example of an MLD Configuration with Traffic Filters

Configuring the Querier

Syntax: [no] ipv6 mld querier

Note: *This command must be issued in a VLAN context.*

This command enables the switch to act as querier on a VLAN.

The **[no]** *form of the command disables the switch from acting as querier on a VLAN.*

The querier function is enabled by default. If another switch or a multicast router is acting as the MLD querier on the VLAN, this switch will defer to that device. If an acting querier stops performing the querier function, all querier-enabled switches and multicast routers on the VLAN will enter an election to determine the next device to act as querier.

For example, to disable the switch from acting as querier on VLAN 8:

```
HP Switch(vlan-8)# no ipv6 mld querier
```

To enable the switch to act as querier on VLAN 8:

```
HP Switch(vlan-8)# ipv6 mld querier
```

Configuring the Query Interval

To specify the number of seconds between membership queries, enter this command with the desired interval.

Syntax: [no] ipv6 mld query-interval <60- 31744>

Note: *This command must be issued in a VLAN context.*
Specifies the number of seconds between membership queries.
The **no** *form of the command sets the interval to the default of 125 seconds.*
Default: 125 seconds..

For example, to set the **query-interval** to 300 seconds on ports in VLAN 8:

```
HP Switch(vlan-8)# ipv6 mld query-interval 300
```

Configuring the Query Maximum Response Time

To specify the maximum amount of time to wait for a response to a query, enter this command.

Syntax: [no] ipv6 mld query-max-response-time<10-128>

> ***Note:*** *This command must be issued in a VLAN context.*
> *Specifies the number of seconds to wait for a response to a query.*
> *The* **no** *form of the command sets the interval to the default of 10 seconds.*
> *Default: 10 seconds.*

For example, to set the **query-max-response-time** to 30 seconds on ports on VLAN 8:

```
HP Switch(vlan-8)# ipv6 mld query-max-response-time 30
```

Configuring the Number of Times to Retry a Query

To specify the number of times to retry a query, enter this command.

Syntax: [no] ipv6 mld robustness <1-8>

> ***Note:*** *This command must be issued in a VLAN context.*
> *Specifies the number of times to retry a query.*
> *The* **no** *form of the command sets the interval to the default of 2.*
> *Default: 2*

For example, to set the number of times to retry a query to 4 on ports on VLAN 8:

```
HP Switch(vlan-8)# ipv6 mld robustness 4
```

Configuring the Last Member Query Interval

You can specify the amount of time that the querier waits to receive a response from members to a group-specific query message by entering this command.

Syntax: [no] ipv6 mld last-member-query-interval <1-2>

> ***Note:*** *This command must be issued in a VLAN context.*
>
> *Sets the amount of time that the querier waits to receive a response from members to a group-specific query message. It also specifies the amount of time between successive group-specific query messages.*
>
> *The* **no** *form of the command sets the interval to the default of 1 second.*
>
> *Default: 1 second.*

For example, to set the amount of time the querier waits to 20 on VLAN 8:

```
HP Switch(vlan-8)# ipv6 mld last-member-query-interval 2
```

Configuring Fast Leave

Syntax: [no] ipv6 mld fastleave <*port-list*>

> *Note: This command must be issued in a VLAN context.*
>
> *This command enables the fast leave function on the specified ports in a VLAN.*
>
> *The* **[no]** *form of the command disables the fast leave function on the specified ports in a VLAN.*
>
> *The fast leave function is enabled by default.*

For example, to disable fast leave on ports in VLAN 8:

```
HP Switch(vlan-8)# no ipv6 mld fastleave 14-15
```

To enable fast leave on ports in VLAN 8:

```
HP Switch(vlan-8)# ipv6 mld fastleave 14-15
```

Configuring Forced Fast Leave

Syntax: [no] ipv6 mld forcedfastleave <*port-list*>

Note: *This command must be issued in a VLAN context.*

This command enables the forced fast leave function on the specified ports in a VLAN.

The **[no]** *form of the command disables the forced fast leave function on the specified ports in a VLAN.*

The forced fast leave function is disabled by default.

For example, to enable forced fast leave on ports in VLAN 8:

```
HP Switch(vlan-8)# ipv6 mld forcedfastleave 19-20
```

To disable forced fast leave on ports in VLAN 8:

```
HP Switch(vlan-8)# no ipv6 mld forcedfastleave 19-20
```

Displaying MLD Status and Configuration

Current MLD Status

Syntax: show ipv6 mld

Displays MLD status information for all VLANs on the switch that have MLD configured.

show ipv6 mld vlan <vid>

Displays MLD status for the specified VLAN
vid—*VLAN ID*

For example, a switch with MLD snooping configured on VLANs 8 and 9 might show the following information:

```
HP Switch# show ipv6 mld

  MLD Service Protocol Info

  Total vlans with MLD enabled                : 2
  Current count of multicast groups joined    : 37

  VLAN ID : 8
  VLAN NAME : VLAN8
  Querier Address : fe80::218:71ff:fec4:2f00 [this switch]
  Querier Up Time : 1h:37m:20s
  Querier Expiry Time : 0h:1m:44s

  Ports with multicast routers :


  Active Group Addresses                   Type ExpiryTime Ports
  ---------------------------------------- ---- ---------- ------------------
  ff02::c                                  FILT 0h:4m:9s   15-21
  ff02::1:2                                FILT 0h:4m:3s   21
  ff02::1:3                                FILT 0h:4m:9s   15-21
  ff02::1:ff00:42                          FILT 0h:4m:0s   19
  ff02::1:ff02:2                           FILT 0h:4m:2s   15
  ff02::1:ff02:3                           FILT 0h:4m:5s   16
  ff02::1:ff03:2                           FILT 0h:4m:2s   17
  ff02::1:ff03:3                           FILT 0h:4m:5s   18
```

Figure 4-4. Example of Displaying the MLD Configuration for All Static VLANs on the Switch

```
 ff02::1:ff04:3                              FILT 0h:4m:5s   20
 ff02::1:ff05:1                              FILT 0h:4m:3s   21
 ff02::1:ff0b:2dfe                           FILT 0h:3m:59s  17
 ff02::1:ff0b:d7d9                           FILT 0h:4m:4s   15
 ff02::1:ff0b:da09                           FILT 0h:4m:5s   18
 ff02::1:ff0b:dc38                           FILT 0h:4m:3s   19
 ff02::1:ff0b:dc8d                           FILT 0h:4m:4s   20
 ff02::1:ff0b:dd56                           FILT 0h:4m:0s   16
 ff02::1:ff12:e0cd                           FILT 0h:4m:5s   21
 ff02::1:ff4e:98a5                           FILT 0h:4m:0s   17
 ff02::1:ff57:21a1                           FILT 0h:3m:58s  20
 ff02::1:ff6b:dd51                           FILT 0h:4m:0s   15
 ff02::1:ff7b:ac55                           FILT 0h:4m:5s   16
 ff02::1:ff8f:61ea                           FILT 0h:4m:1s   19
 ff02::1:ffc8:397b                           FILT 0h:4m:0s   18
 ff3e:30:2001:db8:8:0:7:101                  FILT 0h:4m:4s   15,18,21
 ff3e:30:2001:db8:8:0:7:102                  FILT 0h:4m:13s  16,19
 VLAN ID : 9
 VLAN NAME : VLAN9
 Querier Address : fe80::218:71ff:fec4:2f00 [this switch]
 Querier Up Time : 1h:37m:22s
 Querier Expiry Time : 0h:1m:43s

Ports with multicast routers :

 Active Group Addresses                      Type ExpiryTime Ports
 ----------------------------------------- ---- ---------- --------------------
 ff02::c                                     FILT 0h:4m:12s  3,5,7
 ff02::1:3                                   FILT 0h:4m:12s  3,5,7
 ff02::1:ff02:4                              FILT 0h:4m:4s   3
 ff02::1:ff03:4                              FILT 0h:3m:59s  5
 ff02::1:ff04:4                              FILT 0h:4m:12s  7
 ff02::1:ff0b:dc64                           FILT 0h:4m:0s   7
 ff02::1:ff0b:dcf3                           FILT 0h:4m:2s   3
 ff02::1:ff0b:dd5c                           FILT 0h:4m:4s   5
 ff02::1:ff34:a69e                           FILT 0h:4m:1s   5
 ff02::1:ff8e:11d5                           FILT 0h:3m:57s  7
 ff02::1:ffea:2c4f                           FILT 0h:3m:58s  3
 ff3e:30:2001:db8:9:0:7:111                  FILT 0h:4m:3s   3,5
```

Figure 4-5. Continuation of Figure 4-4

The following information is shown for each VLAN that has MLD snooping enabled:

- VLAN ID number and name
- Querier address: IPv6 address of the device acting as querier for the VLAN
- Querier up time: the length of time in seconds that the querier has been acting as querier
- Querier expiry time: If this switch is the querier, this is the amount of time until the switch sends the next general query. If this switch is not the querier, this is the amount of time in seconds until the current querier is considered inactive (after which a new querier election is held).
- Ports with multicast routers: ports on the VLAN that lead toward multicast routers (if any)
- Multicast group address information for each active group on the VLAN, including:
  - the multicast group address
  - the type of tracking for multicast joins: standard or filtered. If MLD snooping is enabled, port-level tracking results in filtered groups. If MLD snooping is not enabled, joins result in standard groups being tracked by this device. In addition, if hardware resources for multicast filtering are exhausted, new joins may result in standard groups even though MLD snooping is enabled.
  - expiry time: the time until the group expires if no joins are seen
  - the ports that have joined the multicast group

The group addresses you see listed typically result from several network functions. In our example, several of the addresses at the top of the list for each VLAN are IANA well known addresses (see www.iana.org/assignments/ipv6-multicast-addresses); the addresses in the form of ff02::1:ffxx:xxxx are solicited-node multicast addresses (used in IPv6 Neighbor Discovery); and the addresses beginning with ff3e are group addresses used by listeners to streaming video feeds.

Current MLD Configuration

Syntax: show ipv6 mld config

Displays current global MLD configuration for all MLD-enabled VLANS on the switch.

show ipv6 vlan *<vid>* config

Displays current MLD configuration for the specified VLAN, including per-port configuration information.
vid—*VLAN ID*

For example, the general form of the command might look like this:

```
HP Switch# show ipv6 mld config

 MLD Service Config

  Control unknown multicast  [Yes] : Yes
  Forced fast leave timeout  [4] : 4

  VLAN ID VLAN NAME       MLD Enabled Querier Allowed
  ------- --------------- ----------- ---------------
  8       VLAN8           Yes         Yes
  9       VLAN9           Yes         Yes
```

Figure 4-6. Example of a Global MLD Configuration

The following information, for all MLD-enabled VLANs, is shown:

- Control unknown multicast: If this is set to YES, any IPv6 multicast packets that are not joined by an MLD host will be sent only to ports that have detected a multicast router or ports that are administratively forwarded. If this is set to NO (or if MLD snooping is disabled), unjoined IPv6 multicast packets will be flooded out all ports in the VLAN.
- Forced fast leave timeout: the interval between an address specific query and a forced fast leave (assuming no response), in tenths of seconds
- For each VLAN that has MLD enabled:
  - VLAN ID and name
  - whether MLD is enabled on the VLAN (default NO, but the VLAN will not show up on this list unless MLD is enabled)
  - whether the switch can act as querier for the VLAN (default YES)

The specific form of the command might look like this:

```
HP Switch# show ipv6 mld vlan 8 config

 MLD Service Vlan Config

  VLAN ID : 8
  VLAN NAME : VLAN8
  MLD Enabled [No] : Yes
  Querier Allowed [Yes] : Yes

  Port Type      | Port Mode Forced Fast Leave Fast Leave
  ---- --------- + --------- ----------------- ----------
  13   100/1000T | auto      No                Yes
  14   100/1000T | auto      No                Yes
  15   100/1000T | auto      No                Yes
  16   100/1000T | auto      No                Yes
  17   100/1000T | auto      No                Yes
  18   100/1000T | auto      No                Yes
  19   100/1000T | auto      No                Yes
  20   100/1000T | auto      No                Yes
  21   100/1000T | auto      No                Yes
  22   100/1000T | auto      No                Yes
  23   100/1000T | auto      No                Yes
  24   100/1000T | auto      No                Yes
```

Figure 4-7. Example of an MLD Configuration for a Specific VLAN

The following information is shown, if the specified VLAN is MLD-enabled:

- VLAN ID and name
- whether MLD is enabled on the VLAN (default NO, but the information for this VLAN will be listed only if MLD is enabled)
- whether the switch is allowed to act as querier on the VLAN

Ports Currently Joined

Syntax: show ipv6 mld vlan *<vid>* group

Lists the ports currently joined for all IPv6 multicast group addresses in the specified VLAN
vid—*VLAN ID*

show ipv6 mld vlan *<vid>* group *<ipv6-addr>*

Lists the ports currently joined for the specified IPv6 multicast group address in the specified VLAN
vid—*VLAN ID*
ipv6-addr—*address of the IPv6 multicast group for which you want information*

For example, the general form of the command is shown below. The specific form the the command is similar, except that it lists the port information for only the specified group.

```
HP Switch# show ipv6 mld vlan 9 group

 MLD Service Protocol Group Info

 VLAN ID : 9
 VLAN Name : VLAN9

  Filtered Group Address : ff02::c
 Last Reporter : fe80::7061:4b38:dbea:2c4f
  ExpiryTime : 0h:2m:19s

  Port Port Type | Port Mode ExpiryTime
  ---- --------- + --------- --------------------
  3    100/1000T | auto      0h:2m:19s
  5    100/1000T | auto      0h:2m:18s

    .
    .
    .

 Filtered Group Address : ff3e:30:2001:db8:9:0:7:111
  Last Reporter : fe80::7061:4b38:dbea:2c4f
  ExpiryTime : 0h:4m:14s

  Port Port Type | Port Mode ExpiryTime
  ---- --------- + --------- --------------------
  3    100/1000T | auto      0h:4m:14s
  5    100/1000T | auto      0h:4m:09s
```

Figure 4-8. Example of Ports Joined to Multicast Groups in a Specific VLAN

The following information is shown:

- VLAN ID and name
- port information for each IPv6 multicast group address in the VLAN (general group command) or for the specified IPv6 multicast group address (specific group command):
  - group multicast address
  - last reporter: last MLD host to send a join to the group address
  - group expiry time: the time until the group expires if no further joins are seen
  - port name for each port
  - port type for each port: Ethernet connection type
  - port mode for each port: auto (follows MLD snooping rules; that is, packets are forwarded for joined groups), forward (all multicast packets are forwarded to this group), or blocked (all multicast packets are dropped, except that packets for well-known addresses are forwarded)
  - expiry time for each port: amount of time until this port is aged out of the multicast address group, unless a join is received

Statistics

Syntax: show ipv6 mld statistics

Shows MLD statistics for all MLD-enabled VLANs

Syntax: show ipv6 mld vlan <vid> statistics

Shows MLD statistics for the specified VLAN
vid—*VLAN ID*

The general form the of the command shows the total number of MLD-enabled VLANs and a count of multicast groups currently joined. Both forms of the command show VLAN IDs and names, as well as the number of filtered and standard multicast groups and the total number of multicast groups.

For example, the general form of the command:

```
HP Switch# show ipv6 mld statistics

 MLD Service Statistics

  Total vlans with MLD enabled               : 2
  Current count of multicast groups joined   : 36


 MLD Joined Groups Statistics

  VLAN ID VLAN NAME    filtered     standard     total
  ------- ------------ ------------ ------------ ------------
  8       VLAN8        26           0            26
  9       VLAN9        10           0            10
```

Figure 4-9. Example of MLD Statistics for All VLANs Configured

And the specific form of the command:

```
HP Switch# show ipv6 mld vlan 8 statistics

 MLD Statistics

  VLAN ID : 8
  VLAN NAME : VLAN8

  Number of Filtered Groups     : 26
  Number of Standard Groups     : 0
  Total Multicast Groups Joined : 26
```

Figure 4-10. Example of MLD Statistics for a Single VLAN

Counters

Syntax: show ipv6 mld vlan *<vid>* counters

Displays MLD counters for the specified VLAN
vid—*VLAN ID*

```
HP Switch# show ipv6 mld vlan 8 counters

 MLD Service Vlan Counters

  VLAN ID : 8
  VLAN NAME : VLAN8

     General Query Rx                   : 2
     General Query Tx                   : 0
     Group Specific Query Rx            : 0
     Group Specific Query Tx            : 0
     V1 Member Report Rx                : 1589
     V2 Member Report Rx                : 15
     Leave Rx                           : 30
     Unknown MLD Type Rx                : 0
     Unknown Pkt Rx                     : 0
     Forward to Routers Tx Counter      : 83
     Forward to Vlan Tx Counter         : 48
     Port Fast Leave Counter            : 4
     Port Forced Fast Leave Counter     : 0
     Port Membership Timeout Counter   : 28
```

Figure 4-11. Example of MLD Counters for a Single VLAN

The following information is shown:

- VLAN number and name
- For each VLAN:
  - number of general queries received
  - number of general queries sent
  - number of group-specific queries received
  - number of group-specific queries sent
  - number of MLD version 1 member reports (joins) received
  - number of MLD version 2 member reports (joins) received
  - number of leaves received
  - number of MLD packets of unknown type received
  - number of packets of unknown type received
  - number of packets forwarded to routers on this VLAN
  - number of times a packet has been forwarded to all ports on this VLAN
  - number of fast leaves that have occurred
  - number of forced fast leaves that have occurred
  - number of times a join has timed out on this VLAN

Access Control Lists (ACLs)

Introduction

Feature	Default	Menu	CLI	Web
Numbered ACLs				
Standard ACLs	None	—	5-29	—
Extended ACLs	None	—	5-29	—
Named ACLs		—	5-29	—
Enable or Disable an ACL		—	5-39	—
Display ACL Data	n/a	—	5-59	—
Delete an ACL	n/a	—	5-36	—
Configure an ACL from a TFTP Server	n/a	—	5-68	—
Enable ACL Logging	n/a	—	5-71	—

Note

IPv6 ACLs are not supported on HP switches J9779A, J9780A, J9782A, and J9783A.

ACL Applications

ACLs can filter traffic from a host, a group of hosts, or from entire subnets. Where it is necessary to apply ACLs to filter traffic from outside a network or subnet, applying ACLs at the edge of the network or subnet removes unwanted traffic as soon as possible, and thus helps to improve system performance. ACLs filter inbound traffic only and can rapidly consume switch resources. For these reasons, the best places to apply ACLs are on “edge” ports where ACLs are likely to be less complex and resource-intensive.

Optional Network Management Applications

ACLs through a RADIUS server can also be augmented using the Identity-Driven Management (IDM) application available for use with IMC. However, the features described in this chapter can be used without IMC or IDM support, if desired. RADIUS ACLs and VLAN ACLs cannot be used at the same time.

Optional IMC and IDM Applications

HP ICM is a Windows-based network management solution for all manageable HP devices. It provides network mapping and polling capabilities, device auto-discovery and topology, tools for device configuration and management, monitoring network traffic, and alerts and troubleshooting information for HP networks.

Identity Driven Manager (IDM) is an add-on module to the IMC application. IDM extends the functionality of IMC to include authorization control features for edge devices in networks using RADIUS servers and Web-Authentication, MAC-Authentication, or 802.1X security protocols.

For more information, including electronic copies of the IMC and IDM manuals, visit the HP Networking Web site at **www.hp.com/networking/support**. (The IMC and IDM documentation is available under **Network Management** on the **Product manuals page** of the **Technical Support** area.)

General Application Options

Layer 3 IP filtering with Access Control Lists (ACLs) enables you to improve network performance and restrict network use by creating policies for:

- **Switch Management Access:** Permits or denies in-band management access. This includes preventing the use of certain TCP or UDP applications (such as Telnet, SSH, web browser, and SNMP) for transactions between specific source and destination IP addresses.
- **Application Access Security:** Eliminates inbound, unwanted IP, TCP, or UDP traffic by filtering packets where they enter the switch on specific physical ports or trunks.

This chapter describes how to configure, apply, and edit ACLs, and how to monitor the results of ACL actions.

Notes

ACLs can enhance network security by blocking selected IP traffic, and can serve as one aspect of maintaining network security. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

ACLs do not screen non-IP traffic such as AppleTalk and IPX.

For ACL filtering to take effect, configure an ACL and then assign it to the inbound traffic on a statically configured port or trunk.

Table 5-1. Comprehensive Command Summary

Action	Command	Page
Configuring Standard (Numbered) ACLs	HP Switch(config)# [no] access-list < 1-99 > < deny | permit > < any | host <*src-ip-addr* > | *src-ip-address/mask* > [1] [log][2]	5-29
Configuring Extended (Numbered) ACLs	HP Switch(config)# [no] access-list <100-199> < deny | permit > ip < any | host <*src-ip-addr* > | *src-ip-address*/mask > [1] [log][2]	5-29
	HP Switch(config)# [no] access-list < 100-199 > < deny | permit > < tcp | udp > < any | host <*src-ip-addr* > | *src-ip-address*/mask > [1] [eq < *src-port tcp/udp-id* >] < any | host <*dest-ip-addr* > | *dest-ip-address*/mask > [1] [eq < *dest-port tcp/udp-id* >] [log][2]	5-29
Configuring Standard (Named) ACLs	HP Switch(config)# [no] ip access-list standard < *name-str* | 1-99 >	5-40
	HP Switch(config-std-nacl)# < deny | permit > < any | host <*src-ip-addr* > | *src-ip-address/mask* > [1] [log] [2]	5-40
Configuring Extended (Named) ACLs	HP Switch(config)# [no] ip access-list extended < *name-str* | 100-199 >	5-40
	HP Switch(config-std-nacl)# < deny | permit > ip < any | host <*src-ip-addr* > | *src-ip-address*/mask > [1] < any | host <*dest-ip-addr* > | *dest-ip-address*/mask > [1] [log][2]	5-40
	HP Switch(config-std-nacl)# < deny | permit > < tcp | udp > < any | host <*src-ip-addr* > | *src-ip-address*/mask > [1] [eq < *tcp/udp-port-#* | *well-known-port-name* >] < any | host <*dest-ip-addr* > | *dest-ip-address*/mask > [1] [eq < *tcp/udp-port-#* | *well-known-port-name* >] [log][2]	5-40
Enabling or Disabling an ACL	HP Switch(config)# [no] interface < *port-list* > access-group < *name-str* | 1-99 | 100-199 >	5-40
Deleting an ACL from the Switch	HP Switch(config)# no ip access-list < standard < *name-str* | 1-99 >> HP Switch(config)# no ip access-list < extended < *name-str* | 100 -199 >>	5-46

Configuring Extended (Named) ACLs	HP Switch(config)# [no] ip access-list extended < *name-str* | 100-199 >	5-40
	HP Switch(config-std-nacl)# < deny | permit > ip < any | host <*src-ip-addr* > | *src-ip-address*/mask >[1] < any | host <*dest-ip-addr*> | *dest-ip-address*/mask >[1] [log][2]	5-40
	HP Switch(config-std-nacl)# < deny | permit > < tcp | udp > < any | host <*src-ip-addr* > | *src-ip-address*/mask > [1] [eq < *tcp/udp-port-#* | *well-known-port-name* >] < any | host <*dest-ip-addr* > | *dest-ip-address*/mask > [1] [eq < *tcp/udp-port-#* | *well-known-port-name* >] [log][2]	5-40
Enabling or Disabling an ACL	HP Switch(config)# [no] interface < *port-list* > access-group < *name-str* | 1-99 | 100-199 >	5-40
Deleting an ACL from the Switch	HP Switch(config)# no ip access-list < standard < *name-str* | 1-99 >> HP Switch(config)# no ip access-list < extended < *name-str* | 100 -199 >>	5-46

Terminology

Access Control Entry (ACE): An ACE is a policy consisting of criteria and an action to take (permit or deny) on a packet if it meets the criteria. The elements composing the criteria include:

- Source IP address and mask (standard and extended ACLs)

- Destination IP address and mask (extended ACLs only)
- TCP or UDP application port numbers (optional, extended ACLs only)

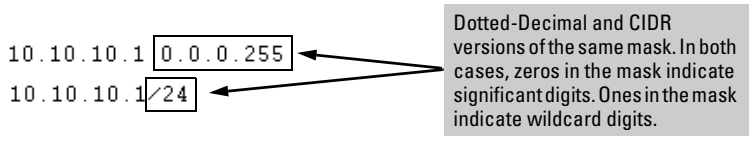
Access Control List (ACL): A list (or set) consisting of one or more explicitly configured Access Control Entries (ACEs) and terminating with an implicit "deny" default which drops any packets that do not have a match with any explicit ACE in the named ACL. The two classes of ACLs are "standard" and "extended". See "Standard ACL" and "Extended ACL".

ACE: See "Access Control Entry".

ACL: See "Access Control List".

ACL ID: A number or alphanumeric string used to identify an ACL. A *standard* ACL ID can have either a number from 1 to 99 or an alphanumeric string. An *extended* ACL ID can have either a number from 100 to 199 or an alphanumeric string.

ACL Mask: Follows an IP address (source or destination) listed in an ACE to specify either a subnet or a group of devices. Defines which bits in a packet's corresponding IP addressing must exactly match the IP addressing in the ACE, and which bits need not match (wildcards). For example:



As shown above, zeros in an ACL mask specify an exact match requirement for IP addresses, and ones specify a wildcard. In this example, a matching IP address would be any address in the range 10.10.10.1-255. (See also "How an ACE Uses a Mask To Screen Packets for Matches" on page 5-26.)

DA: The acronym for *Destination IP Address*. In an IP packet, this is the destination IP address carried in the header, and identifies the destination intended by the packet's originator. In an extended ACE, this is the second of two IP addresses required by the ACE to determine whether there is a match between a packet and the ACE. See also "SA".

Deny: An ACE configured with this action causes the switch to drop an inbound packet for which there is a match within an applicable ACL. As an option, you can configure the switch to generate a logging output to a Syslog server and a console session.)

Extended ACL: This type of Access Control List uses layer-3 IP criteria composed of source and destination IP addresses and (optionally) TCP or UDP port criteria to determine whether there is a match with an IP packet.You can apply an extended ACL to inbound traffic on a port or trunk, including any inbound traffic with a DA belonging to the switch itself. Extended ACLs require an identification number (ID) in the range of 100 - 199 or an alphanumeric name.

Implicit Deny: If the switch finds no matches between an inbound packet and the configured criteria in an applicable ACL, then the switch denies (drops) the packet with an implicit "deny IP any" operation. You can preempt the implicit "deny IP any" in a given ACL by configuring **permit any** (standard) or **permit IP any any** (extended) as the last explicit ACE in the ACL. Doing so permits an inbound packet that is not explicitly permitted or denied by other ACEs configured sequentially earlier in the ACL. Unless otherwise noted, "implicit deny IP any" refers to the "deny" action enforced by both standard and extended ACLs.

Inbound Traffic: For the purpose of defining where the switch applies ACLs to filter traffic, inbound traffic is any IP packet that:

- Enters the switch through a physical port.
- Has a destination IP address (DA) that meets either of these criteria:
  - The packet's DA is for an external device.
  - The packet's DA is for an IP address configured on the switch itself. (This increases your options for protecting the switch from unauthorized management access.)

Because ACLs are assigned to physical ports or port trunks, an ACL that filters inbound traffic on a particular port or trunk examines packets meeting the above criteria that enter the switch through that port or trunk.

Outbound Traffic: This is any traffic *leaving the switch* through a physical port or trunk. The switch does not apply ACLs to outbound traffic or internally where routed traffic moves between VLANs. That is, ACL operation is not affected by enabling or disabling routing on the switch. (Refer also to "ACL Inbound Application Points" on page 5-7.)

Permit: An ACE configured with this action allows a port or trunk to permit an inbound packet for which there is a match within an applicable ACL.

SA: The acronym for *Source IP Address*. In an IP packet, this is the source IP address carried in the IP header, and identifies the packet's sender. In an extended ACE, this is the first of two IP addresses used by the ACE to determine whether there is a match between a packet and the ACE. See also "DA".

Standard ACL: This type of Access Control List uses layer-3 IP criteria of source IP address to determine whether there is a match with an inbound IP packet. You can apply a standard ACL to inbound traffic on a port or trunk, including any inbound traffic with a DA belonging to the switch itself. Standard ACLs require an identification number (ID) in the range of 1 - 99 or an alphanumeric name.

VLAN ACL (VACL): An ACL applied to all IPv6 traffic entering the switch on a given VLAN interface. See also “Access Control List”.

Wildcard: The part of a mask that indicates the bits in a packet’s IP addressing that do not need to match the corresponding bits specified in an ACL. See also **ACL Mask** on page 5-5.

Overview

Types of IP ACLs

Standard ACL: Use a standard ACL when you need to permit or deny traffic based on source IP address. Standard ACLs are also useful when you need to quickly control a performance problem by limiting traffic from a subnet, group of devices, or a single device. (This can block all inbound IP traffic from the configured source, but does not block traffic from other sources within the network.) This ACL type uses a numeric ID of 1 through 99 or an alphanumeric ID string. You can specify a single host, a finite group of hosts, or any host.

Extended ACL: Use extended ACLs whenever simple IP source address restrictions do not provide the breadth of traffic selection criteria you want for a port or trunk. Extended ACLs allow use of the following criteria:

- Source and destination IP addresses
- TCP application criteria
- UDP application criteria

ACL Inbound Application Points

You can apply ACL filtering to IP traffic inbound on a physical port or static trunk with a destination (DA):

- On another device. (ACLs are not supported on dynamic LACP trunks.)

- On the switch itself. In figure 5-1, below, this would be any of the IP addresses shown in VLANs "A", "B", and "C" on the switch. (IP routing need not be enabled.)

The switch can apply ACL filtering to traffic *entering the switch* on ports and/or trunks configured to apply ACL filters. For example, in figure 5-1 you would assign an inbound ACL on port 1 to filter a packet from the workstation 10.28.10.5 to the server at 10.28.20.99. Note that all ACL filtering is performed on the inbound port or trunk. Routing may be enabled or disabled on the switch, and any permitted inbound traffic may have any valid destination.

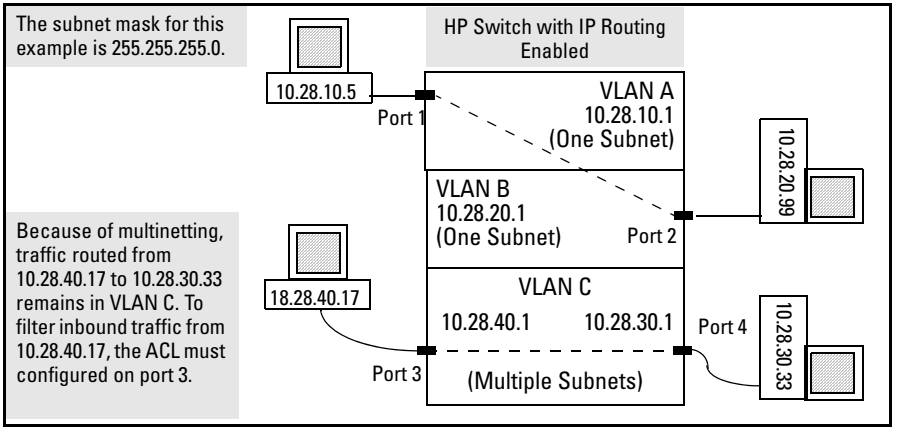


Figure 5-1. Example of Filter Applications

VACL Applications

IPv6 VACLs filter traffic entering the switch on a VLAN configured with the "VLAN" ACL option.

vlan < *vid* > ipv6 access-group < *vacl-identifier* > vlan

For example, in figure 5-2, you would assign a VACL to VLAN 2 to filter all inbound switched or routed IPv6 traffic received from clients on the 2001:db8 :0:222:: network. In this instance, routed IPv6 traffic received on VLAN 2 from VLANs 1 or 3 would not be filtered by the VACL on VLAN 2.

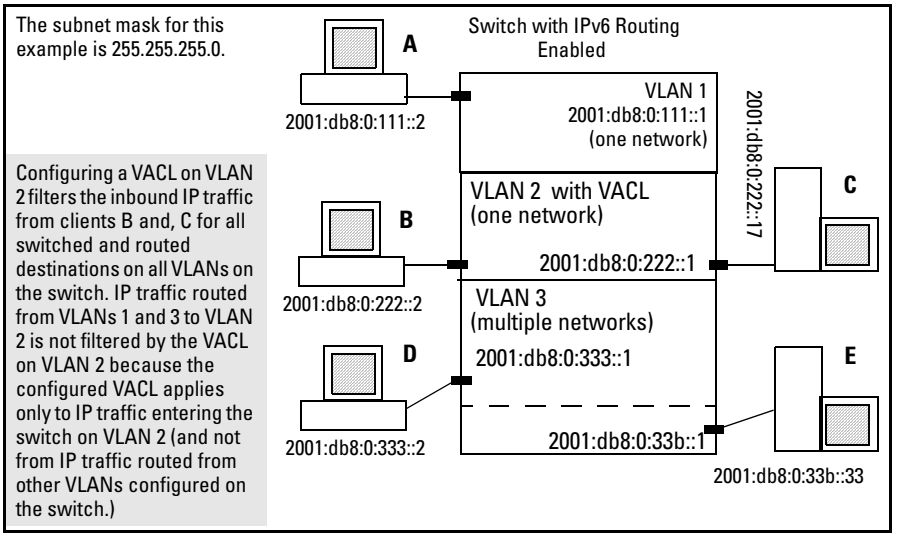


Figure 5-2. Example of VACL Filter Applications on IPv6 Traffic Entering the Switch

Note

The switch allows one IPv6 VACL assignment configured per VLAN. This is in addition to any other IPv6 ACL applications assigned to the IP routing interface or to ports in the VLAN.

Features Common to All ACLs

- On any port or static trunk you can apply one ACL to inbound traffic.
- Any ACL can have multiple entries (ACEs).
- You can apply any one ACL to multiple ports and trunks.
- A source or destination IP address and a mask, together, can define a single host, a range of hosts, or all hosts.
- Before changing the content of an ACL assigned to one or more ports or trunks, you must first remove the ACL from those ports or trunks.
- Every standard ACL includes an implied "**deny any**" as the last entry, and every extended ACL includes an implied "**deny IP any any**" as the last entry. The switch applies this action to any packets that do not match other criteria in the ACL.

- In any ACL, you can apply an ACL log function to ACEs that have a "deny" action. The logging occurs when there is a match on a "deny" ACE. (The switch sends ACL logging output to Syslog and, optionally, to a console session.)
- Standard and Extended ACL features cannot be combined in one ACL.

You can configure ACLs using either the CLI or a text editor. The text-editor method is recommended when you plan to create or modify an ACL that has more entries than you can easily enter or edit using the CLI alone. Refer to "Creating or Editing ACLs Offline" on page 5-68.

General Steps for Planning and Configuring ACLs

1. Identify the traffic type to filter. Options include:
   - Any inbound IP traffic
   - Inbound TCP traffic only
   - Inbound UDP traffic only
2. The SA and/or the DA of inbound traffic you want to permit or deny.
3. Determine the best points at which to apply specific ACL controls. For example, you can improve network performance by filtering unwanted traffic at the edge of the network instead of in the core.
4. Design the ACLs for the selected control points. Where you are using explicit "deny" ACEs, you can optionally use the ACL logging feature to help verify that the switch is denying unwanted packets where intended. Remember that excessive ACL logging activity can degrade the switch's performance. (Refer to "Enable IPv6 ACL "Deny" Logging" on page 5-71.)
5. Create the ACLs in the selected switches.
6. Assign the ACLs to filter the inbound traffic on ports and/or static trunk interfaces configured on the switch.
7. Test for desired results.

For more details on ACL planning considerations, refer to "Planning an ACL Application" on page 5-17.

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to use the **no ip source-route** command to disable source routing on the switch. (If source routing is disabled in the running-config file, the **show running** command includes "**no ip source-route**" in the running-config file listing.)

ACL Operation

Introduction

An ACL is a list of one or more Access Control Entries (ACEs), where each ACE consists of a matching criteria and an action (permit or deny). An ACL applies only to the switch in which it is configured. ACLs operate on assigned ports and static trunks, and filter these traffic types:

- Traffic entering the switch. (Note that ACLs do not screen traffic at any internal point.where traffic moves between VLANs or subnets within the switch; only on inbound ports and static trunks. Refer to "ACL Inbound Application Points" on page 5-7.)
- Switched or routed traffic entering the switch and having an IP address on the switch as the destination

You can apply one inbound ACL to each port and static trunk configured on the switch. The complete range of options includes:

- **No ACL** assigned. (In this case, all traffic entering the switch on the interface does so without any ACL filtering, which is the default.)
- **One ACL** assigned to filter the inbound traffic entering the switch on the interface.
- **Multiple Assignments for the same ACL.** (The switch allows one ACL assignment to an interface, but you can assign the same ACL to multiple interfaces.)

Note

On a given port or trunk, after you assign an ACL, the default action is to deny any traffic that is not specifically permitted by the ACL. (This applies only to the inbound traffic flow filtered by the ACL.)

The Packet-Filtering Process

Sequential Comparison and Action. When the switch uses an ACL to filter a packet, it sequentially compares each ACE's filtering criteria to the corresponding data in the packet until it finds a match.

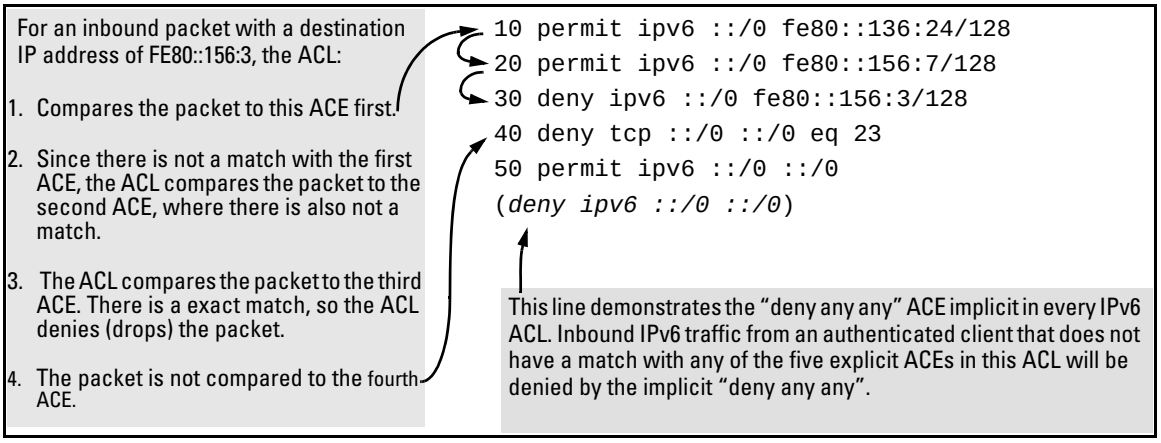


Figure 5-3. Example of Sequential Comparison

As shown above, the ACL tries to apply the first ACE in the list. If there is not a match, it tries the second ACE, and so on. When a match is found, the ACL invokes the configured action for that entry (permit or drop the packet) and no further comparisons of the packet are made with the remaining ACEs in the list. This means that when an ACE whose criteria matches a packet is found, the action configured for that ACE is invoked, and any remaining ACEs in the ACL are ignored. *Because of this sequential processing, successfully implementing an ACL depends in part on configuring ACEs in the correct order for the overall policy you want the ACL to enforce.*

Implicit Deny. If a packet does not have a match with the criteria in any of the ACEs in the ACL, the switch denies (drops) the packet. (This is termed *implicit deny.*) If you need to override the implicit deny so that any packet that does not have a match will be permitted, then you can enter **permit any** as the last ACE in the ACL. This directs the switch to permit (forward) any packets that do not have a match with any earlier ACE listed in the ACL, and prevents these packets from being filtered by the implicit deny.

Note on Implicit Deny

For ACLs configured to filter inbound packets, note that Implicit Deny filters *any packets, including those with a DA specifying the switch itself*. This operation helps to prevent management access from unauthorized IP sources.

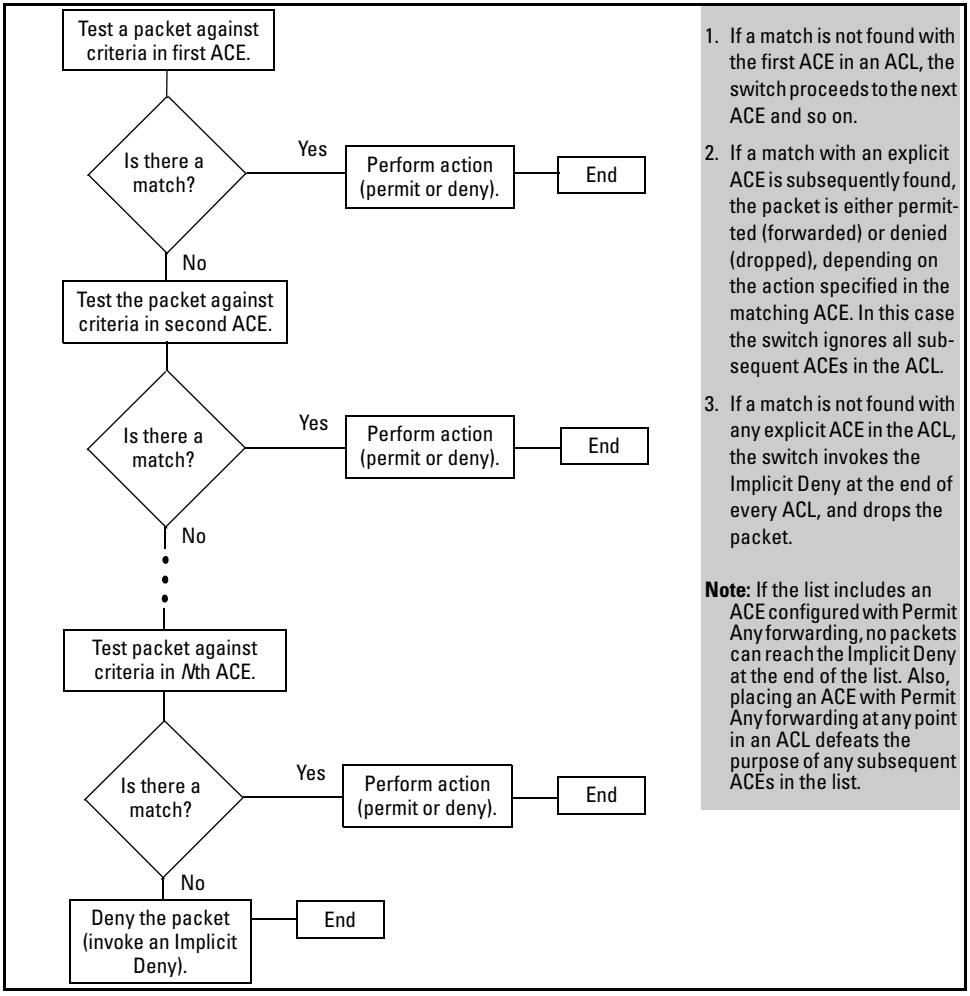


Figure 5-4. The Packet-Filtering Process in an ACL with *N* Entries (ACEs)

Note

The order in which an ACE occurs in an ACL is significant. For example, if an ACL contains six ACEs, but the first ACE allows "Permit Any" forwarding, then the ACL permits all IPv6 traffic, and the remaining ACEs in the list do not apply, even if they have a match with any traffic permitted by the first ACE.

For example, suppose you want to configure an ACL (with an ID of "Test-02") to invoke these policies for IPv6 traffic entering the switch on VLAN 100:

1. Permit inbound IPv6 traffic from 2001:db8:0:fb::11:42.
2. Deny only the inbound Telnet traffic from 2001:db8:0:fb::11:101.
3. Permit inbound IPv6 traffic from 2001:db8:0:fb::11:101.
4. Permit only inbound Telnet traffic from 2001:db8:0:fb::11:33.
5. Deny any other inbound IPv6 traffic.

The following ACL, when assigned to filter inbound traffic on VLAN 100, supports the above case:

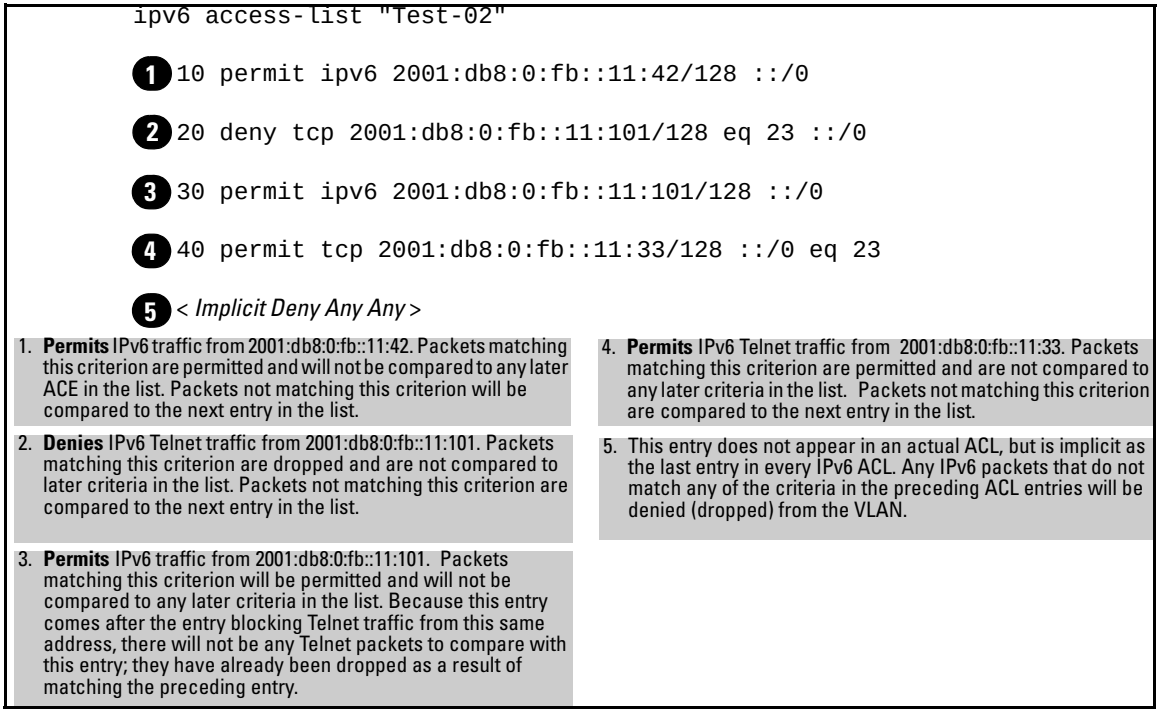


Figure 5-5. Example of How an ACL Filters Packets

To assign the above ACL, you would use this command:

```
HP Switch(config)# vlan 100 ipv6 access-group Test-02 vlan
```

For example, suppose you want to configure an ACL on the switch (with an ID of "Test-02") to invoke these policies for IPv6 traffic entering the switch on VLAN 12:

1. Permit inbound IPv6 traffic from 2001:db8:0:fb::11:42.
2. Deny only the inbound Telnet traffic from 2001:db8:0:fb::11:101.Permit inbound IPv6 traffic from 2001:db8:0:fb::11:101.
3. Permit only inbound Telnet traffic from 2001:db8:0:fb::11:33.
4. Deny other inbound IPv6 traffic.

The following ACL model, when assigned to inbound filtering on an interface, supports the above case:

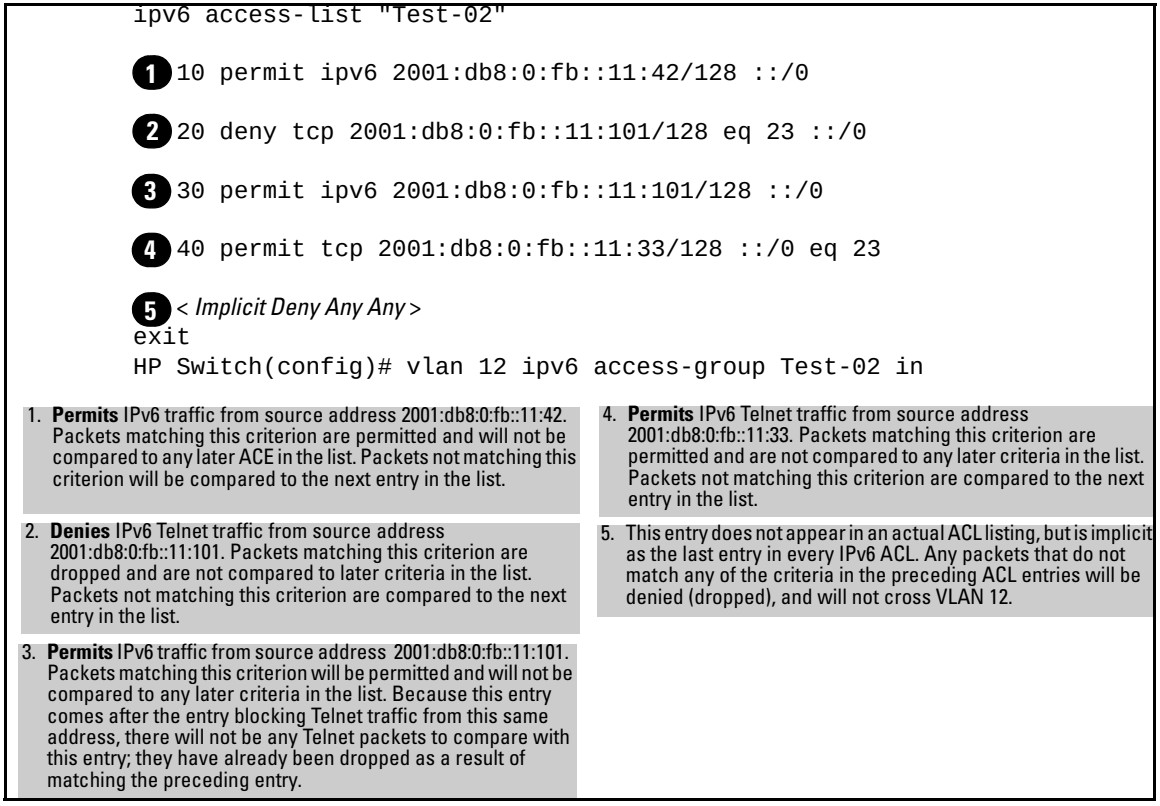


Figure 5-6. Example of How an ACL Filters Packets

It is important to remember that ACLs configurable on the switch include an implicit **deny ipv6 any any**. That is, IPv6 packets that the ACL does not *explicitly* permit or deny will be *implicitly* denied, and therefore dropped instead of forwarded on the interface. If you want to preempt the implicit deny so that packets not explicitly denied by other ACEs in the ACL will be permitted, insert an explicit **permit ipv6 any any** as the last ACE in the ACL. Doing so permits any packet not explicitly denied by earlier entries. (Note that this solution would not apply in the preceding example, where the intention is for the switch to forward only the explicitly permitted packets entering the switch on VLAN 100.) (Note that this solution does not apply in the preceding example, where the intention is for the switch to forward only explicitly permitted packets routed on VLAN 12.)

Planning an ACL Application

Before creating and implementing ACLs, you should understand the switch resources available to support ACL operation, define the policies you want your ACLs to enforce, and understand how your ACLs will impact your network users.

Switch Resource Usage

ACLs load resources in ways that require more careful attention to resource usage when planning a configuration using these features. Otherwise, there is an increased possibility of fully consuming some resources, which means that at some point the switch would not support further ACL configurations. This section describes resource planning for ACLs on your switch.

Prioritizing and Monitoring ACL and QoS, Feature Usage

If you want to configure ACLs on your switch, plan and implement your configuration in descending order of feature importance. This will help to ensure that the most important features are configured first. Also, if insufficient resources become a problem, this approach can help you recognize how to distribute the desired feature implementations across multiple switches to achieve your objectives.

ACL Resource Usage and Monitoring

ACL configurations use internal rules on a per-device basis. There are 128 rules available for configuring ACLs with the CLI and 128 rules available for configuring ACLs with IDM. You can apply a CLI ACL and and IDM ACL on the same port at the same time.

The switch uses resources required by the ACEs in an ACL when you apply the ACL to one or more port and/or static trunk interfaces.

Rule Usage

- There is only one implicit “deny any” entry per device for CLI ACLs, and one implicit “deny any” entry per device for IDM ACLs.
- The implicit “deny any” entry is created only the first time an ACL is applied to a port. After that the port-map is updated for that “deny any” entry to include or remove additional ports.
- Each ACE, including the implicit **deny any** ACE in a standard ACL, uses one rule.
- There is a separate rule for every ACE whether the ACE uses the same mask or a new mask.
- Two hardware rules are used for any “permit” ACE with TCP or UDP specified. One rule is for normal packets and one is for fragmented packets.

Table 5-2 on page 5-18 summarizes switch use of resources to support ACES.

Table 5-2. ACL Rule and Mask Resource Usage

ACE Type	Rule Usage
Standard ACLs	
Implicit **deny any** (automatically included in any standard ACL, but not displayed by **show access-list** < *acl-#* > command).	1
First ACE entered	1
Next ACE entered with same ACL mask	1
Next ACE entered with a different ACL mask	1
Closing ACL with a deny any or permit any ACE having the same ACL mask as the preceding ACE	1
Closing ACL with a deny any or permit any ACE having a different ACL mask than the preceding ACE	1
Extended ACLs	
Implicit **deny ip any** (automatically included in any standard ACL, but not displayed by **show access-list** < *acl-#* > command).	1

ACE Type	Rule Usage
First ACE entered	1
Next ACE entered with same SA/DA ACL mask and same IP or TCP/UDP protocols specified	2
Next ACE entered with any of the following differences from preceding ACE in the list: – Different SA or DA ACL mask – Different protocol (IP as opposed to TCP/UDP) specified in either the SA or DA	1
Closing an ACL with a **deny ip any any** or **permit ip any any** ACE preceded by an IP ACE with the same SA and DA ACL masks	1
Closing an ACL with a **deny ip any any** or **permit ip any any** ACE preceded by an IP ACE with different SA and/or DA ACL masks	1

The following two CLI commands are useful for planning and monitoring rule and mask usage in an ACL configuration.

Syntax: access-list resources help

Provides a quick reference on how ACLs use rule resources. Includes most of the information in table 5-2, plus an ACL usage summary.

Syntax: show access-list resources

Shows the number of rules used, maximum rules available, resources used and resources required for ACLs created with Identity Manager (IDM) and for ACLs created with the CLI.

Managing ACL Resource Consumption

As shown in table 5-2, changes in IP subnet masks or changes in IP or TCP/UDP applications among consecutive ACEs in an assigned ACL can rapidly consume resources. Adding a new ACE to an ACL consumes one rule. An extensive ACL configuration can fully subscribe the 128 rule resources available on the switch.

Oversubscribing Available Resources

If a given ACL requires more rule resources than are available, then the switch cannot apply the ACL to *any* of the interfaces specified for that ACL. In this case, the **access-group** command fails and the CLI displays the following:

- In the CLI:

  ```
  Unable to apply access control list.
  ```

- In the Event Log (and in a Syslog server, if configured on the switch):

`ACL: unable to apply ACL <acl-#> to port <port-#>, failed to add entry < # >`

(Note that < *port-#*> is the first port in the assignment command that was unable to support the ACL.)

Troubleshooting a Shortage of Resources

Do the following to determine how to change resource usage to allow the ACL you want to configure:

1. Use the **show access-list resources** command
2. Use **show** commands to identify the currently configured ACL policies.
3. Determine which of the existing policies you can remove to free up rule resources for the ACL policy you want to implement. Depending on your network topology and configuration, you can free up rule resources by moving some policies to other devices. Another alternative is to inspect the switch's existing configuration for inefficient applications that could be removed or revised to achieve the desired policies with less resource usage. Table 5-2 on page 5-18 and the information displayed by the **access-list resources help** command, can help you to determine the resource usage of ACL policies.

Example of ACL Resource Usage

This example illustrates how to check for current rule availability, and then how to create and assign an ACL, and then to verify its effect on rule resources. (For more detailed information on configuring and applying ACLs, refer to the later sections of this chapter.)

Viewing the Current Rule Usage

The **show access-list resources** command displays current information about rules and resources.

```
HP Switch(config)# show access-list resources
ACL Resource Usage

                  Rules Rules   Resources Resources
 Feature          Used  Maximum Used      Required
 -----------------|-----|-------|---------|--------
 cli-acl          |  15 |  128  |       1 |      1
 idm-acl          |  0  |  128  |       0 |      2
```

Figure 5-7. Example of Rules Used and Resources Used and Required

Standard ACL Using a Subset of the Switch's Ports. Suppose that ports 1 - 4 belong to the following VLANs:

- VLAN 1: 10.10.10.1
- VLAN 2: 10.10.11.1
- VLAN 3: 10.10.12.1

(Assume that ports 1-4 are tagged members of VLAN 22, although tagged/untagged ports do not affect ACL operation because ACLs examine all inbound traffic, regardless of VLAN membership.)

The system administrator wants to:

- Permit inbound VLAN 1 traffic on all ports
- Permit inbound VLAN 2 traffic on ports 1 - 4 from hosts 10.10.10.1-30
- Deny inbound VLAN 2 traffic on ports 1 - 4 from hosts 10.10.10.31-255
- Permit inbound VLAN 3 traffic on all ports.

Because all ports in the example have the same inbound traffic requirements for ACL filtering, the system administrator needs to create only one ACL for application to all four ports.

- All inbound 10.10.10.*x* (VLAN 1) traffic is allowed on all ports.
- For the inbound 10.10.11.*x* (VLAN 2) traffic, the fourth octet of the ACL mask includes an overlap of permit and deny use on the "16" bit, which will require two different ACEs in the ACL. That is:
  - To deny hosts in the range of 31-255 in the fourth octet, it is necessary to use an ACE that specifies the leftmost four bits of the octet.
  - To permit hosts in the range of 1-30 in the fourth octet, it is necessary to use and ACE that specifies the rightmost five bits of the octet.

  The overlap[1] can be illustrated as shown here:

Bit Values in the Fourth Octet	128	64	32	16	8	4	2	1
Bits Needed To Deny Hosts 31 - 255 (4th Octet Mask: 0.0.0.224)								
Bits Needed To Permit Hosts 1 - 30 (4th Octet Mask: 0.0.0.31)								

[1] For more information,see "Using CIDR Notation To Enter the IPv6 ACL Prefix Length" on page 5-36.

The overlap on the "16" bit means that it is necessary for the ACL to deny the host at 10.10.11.31 before permitting the hosts in the range of 10.10.10.1 - 30. The complete sequence is:

1. Permit all inbound traffic from 10.10.10.*x*.
2. Permit all inbound traffic from 10.10.12.*x*.
3. Deny the host at 10.10.11.31.
4. Permit the hosts in the range of 10.10.11.1 - 30.
5. Allow the implicit deny (automatically present in all ACLs) to deny all other traffic, which will automatically include the hosts in the range 10.10.10.32 - 255.

```
HP Switch(config)# access-list 1 permit 10.10.10.1/24
HP Switch(config)# access-list 1 permit 10.10.12.1/24
HP Switch(config)# access-list 1 deny host 10.10.11.31
HP Switch(config)# access-list 1 permit 10.10.11.1/27
HP Switch(config)# show access-list 1

Access Control Lists

  Name: 1
  Type: Standard
  Applied: No

 ID  action       IP              Mask            Log
-----------------------------------------------------
  1  permit  std  10.10.10.1      0.0.0.255
  2  permit  std  10.10.12.1      0.0.0.255
  3    deny  std  10.10.11.31     0.0.0.0
  4  permit  std  10.10.11.1      0.0.0.31

HP Switch(config)# interface 1-4 access-group 1 in
```

Every standard ACL has at least two ACEs; the first ACE that you configure, and the implicit **deny any** ACE that follows all other configured ACEs in the ACL..

Figure 5-8. Example of Configuring an ACL

Traffic Management and Improved Network Performance

You can use ACLs to block unnecessary traffic caused by individual hosts, workgroups, or subnets, and to block user access to subnets, devices, and services. Answering the following questions can help you to design and properly position ACLs for optimum network usage.

- What are the logical points for minimizing unwanted traffic? In many cases it makes sense to block unwanted traffic from the core of your network by configuring ACLs to drop such traffic at or close to the edge of the network. (The earlier in the network path you block unwanted traffic, the greater the benefit for network performance.)
- What traffic should you explicitly block? Depending on your network size and the access requirements of individual hosts, this can involve creating a large number of ACEs in a given ACL (or a large number of ACLs), which increases the complexity of your solution and rapidly consumes the resources.
- What traffic can you implicitly block by taking advantage of the implicit **deny any** to deny traffic that you have not explicitly permitted? This can reduce the number of entries needed in an ACL and make more economical use of switch resources.
- What traffic should you permit? In some cases you will need to explicitly identify permitted traffic. In other cases, depending on your policies, you can insert a **permit any** (standard ACL) or **permit ip any any** (extended ACL) entry at the end of an ACL. This means that all IP traffic not specifically matched by earlier entries in the list will be permitted.

Security

ACLs can enhance security by blocking inbound IP traffic carrying an unauthorized source IP address (SA). This can include:

- Blocking access to or from subnets in your network
- Blocking access to or from the internet
- Blocking access to sensitive data storage or restricted equipment

- Preventing the use of specific TCP or UDP functions (such as Telnet, SSH, web browser) for unauthorized access

You can also enhance switch management security by using ACLs to block inbound IP traffic that has the switch itself as the destination address (DA).

Caution

ACLs can enhance network security by blocking selected IP traffic, and can serve as one aspect of maintaining network security. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

Note

ACLs do not screen non-IP traffic such as AppleTalk and IPX.

Guidelines for Planning the Structure of an ACL

The first step in planning a specific ACL is to determine where you will apply it. (Refer to "ACL Inbound Application Points" on page 5-7.) You must then determine the order in which you want the individual ACEs in the ACL to filter traffic. Some applications require high usage of the resources the switch uses to support ACLs. In these cases it is important to order the individual ACEs in a list to avoid unnecessarily using resources. For more on this topic, refer to "Planning an ACL Application" on page 5-17.

- The first match dictates the action on a packet. possible, subsequent matches are ignored.
- On any ACL, the switch implicitly denies packets that are not explicitly permitted or denied by the ACEs configured in the ACL. If you want the switch to forward a packet for which there is not a match in an ACL, add **permit any** as the last ACE in an ACL. This ensures that no packets reach the implicit **deny any** case.
- Generally, you should list ACEs from the most specific (individual hosts) to the most general (subnets or groups of subnets) unless doing so permits traffic that you want dropped. For example, an ACE allowing a small group of workstations to use a specialized printer should occur earlier in an ACL than an entry used to block widespread access to the same printer.

ACL Configuration and Operating Rules

- **Per-Interface ACL Limits.** At a minimum an ACL will have one explicit "deny" Access Control Entry. You can assign one ACL per interface, as follows:
  - Standard ACLs—Numeric range: 1 - 99
  - Extended ACLs—Numeric range: 100 - 199
  - Named (Extended or Standard) ACLs: Up to the maximum number of ports on the switch (minus any numeric ACL assignments)

- **Implicit "deny any":** In any ACL, the switch automatically applies an implicit "deny IP any" that does not appear in **show** listings. This means that the ACL denies any packet it encounters that does not have a match with an entry in the ACL. Thus, if you want an ACL to permit any packets that you have not expressly denied, you must enter a **permit any** or **permit ip any any** as the last visible ACE in an ACL. Because, for a given packet the switch sequentially applies the ACEs in an ACL until it finds a match, any packet that reaches the **permit any** or **permit ip any any** entry will be permitted, and will not encounter the "deny ip any" ACE the switch automatically includes at the end of the ACL. For an example, refer to figure 5-6 on page 5-16.

- **Explicitly Permitting Any IP Traffic:** Entering a **permit any** or a **permit ip any any** ACE in an ACL permits all IP traffic not previously permitted or denied by that ACL.

- **Explicitly Denying Any IP Traffic:** Entering a **deny any** or a **deny ip any any** ACE in an ACL denies all IP traffic not previously permitted or denied by that ACL.

- **An ACL Assignment Is Exclusive:** The switch allows one ACL assignment on an interface. If a port or static trunk already has an ACL assigned, you cannot assign another ACL to the interface without first removing the currently assigned ACL.

- **Replacing One ACL with Another:** Where an ACL is already assigned to an interface, you must remove the current ACL assignment before assigning another ACL to that interface. If an assignment command fails because one or more interfaces specified in the command already have an ACL assignment, the switch generates this message in the CLI and in the Event Log:

  `<` *acl-list-#* `>: Unable to apply access control list.`

- **ACLs Operate On Ports and Static Trunk Interfaces:** You can assign an ACL to any port and/or any statically configured trunk on the switch. ACLs do not operate with dynamic (LACP) trunks.
- **Before Modifying an Applied ACL, You Must First Remove It from All Assigned Interfaces:** An ACL cannot be changed while it is assigned to an interface.
- **Before Deleting an Applied ACL, You Must First Remove It from All Interfaces to Which It Is Assigned:** An assigned ACL cannot be deleted.
- **Port and Static Trunk Interfaces:**
  - Removing a port from an ACL-assigned trunk returns the port to its default settings.
  - To add a port to a trunk when an ACL is already assigned to the port, you must first remove the ACL assignment from the port.
  - Adding a new port to an ACL-assigned trunk automatically applies the ACL to the new port.

How an ACE Uses a Mask To Screen Packets for Matches

For an IPv6 ACL, a match with a packet occurs when both the protocol and the SA/DA configured in a given ACE within the ACL are a match with the same criteria in a packet being filtered by the ACL.

In IPv6 ACEs, prefixes define how many leading bits in the SA and DA to use for determining a match. That is, the switch uses IPv6 prefixes in CIDR format to specify how many leading bits in a packet's SA and DA must be an exact match with the same bits in an ACE. The bits to the right of the prefix are "wildcards", and are not used to determine a match.

Prefix	Range of Applicable Addresses	Examples
/0	any IPv6 host	::/0
/ 1 — /127	all IPv6 hosts within the range defined by the number of bits in the prefix	2001:db8::/48 2001:db8::/64
/128	one IPv6 host	2001:db8::218:71ff:fec4:2f00/128

For example, the following ACE applies to Telnet packets from a source address where the leading bits are set to 2001:db8:10:1 and any destination address where the leading bits are set to 2001:db8:10:1:218:71ff:fec.

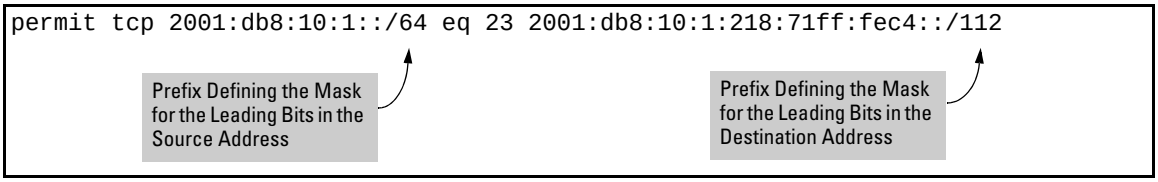


Figure 5-9. Example of SA/DA Prefix Lengths

Thus, in the above example, if an IPv6 telnet packet has an SA match with the ACE's leftmost 64 bits and a DA match with the ACE's leftmost 112 bits, then there is a match and the packet is permitted. In this case, the source and destination addresses allowed are:

Address	Prefix	Range of Unicast Addresses
Source (SA)	2001:db8:10:1	< *prefix* >::0 to < *prefix* >:FFFF:FFFF:FFFF:FFFF
Destination (DA)	2001:db8:10:1:218:71ff:fec4	< *prefix* >:0 to < *prefix* >:FFFF

To summarize, when the switch compares an IPv6 packet to an ACE in an ACL, it uses the subnet prefixes configured with the SA and DA in the ACE to determine how many leftmost, contiguous bits in the ACE's SA and DA must be matched by the same bits in the SA and DA carried by the packet. Thus, the subnet prefixes specified with the SA and DA in an ACE determine the ranges of source and destination addresses acceptable for a match between the ACE and a packet being filtered.

Prefix Usage Differences Between ACLs and Other IPv6 Addressing

For ACLs, the prefix is used to specify the leftmost bits in an address that are meaningful for a packet match. In other IPv6 usage, the prefix separates network and subnet values from the device identifier in an address.

Prefix Usage	Examples	Notes
For an SA or DA in the ACE belonging to an IPv6 ACL, the associated prefix specifies how many consecutive, leading bits in the address are used to define a match with the corresponding bits in the SA or DA of a packet being filtered.	2620:0:a03:e102:215:60ff:fe7a:adc0/128	All bits. Used for a specific SA or DA.
	2620:0:a03:e102:215/80	The first 80 bits. Used for an SA or DA having 2620:0:a03:e102:215 in the leftmost 80 bits of an address.
	::/0	Zero bits. Used to allow a match with "Any" SA or DA.
For the IPv6 address assigned to a given device, the prefix defines the type of address and the network and subnet in which the address resides. In this case, the bits to the right of the prefix comprise the device identifier.	fe80::215:60ff:fe7a:adc0/64	Link-Local address with a prefix of 64 bits and a device ID of 64 bits.
	2620:0:a03:e102:215:60ff:fe7a:adc0/64	Global unicast address with a prefix of 64 bits and a device ID of 64 bits.

Configuring and Assigning an ACL

Overview

General Steps for Implementing ACLs

1. Configure at least one ACL. This creates and stores the ACL in the switch configuration.
2. Assign an ACL. This applies the ACL to the inbound traffic on one or more designated interfaces.

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to disable source routing on the switch. To do so, execute **no ip source-route**.

Types of ACLs

- **Standard ACL:** Uses only a packet's source IP address as a criterion for permitting or denying the packet. For a standard ACL ID, use either a unique numeric string in the range of 1-99 or a unique name string of up to 64 alphanumeric characters.
- **Extended ACL:** Offers the following criteria as options for permitting or denying a packet:
  - Source IP address
  - Destination IP address
  - TCP or UDP criteria

  For an extended ACL ID, use either a unique number in the range of 100-199 or a unique name string of up to 64 alphanumeric characters.

Carefully plan your ACL application before configuring specific ACLs. For more on this topic, refer to "Planning an ACL Application" on page 5-17.

ACL Configuration Structure

After you enter an ACL command, you may want to inspect the resulting configuration. This is especially true where you are entering multiple ACEs into an ACL. Also, it will be helpful to understand the configuration structure when using later sections in this chapter.

The basic ACL structure includes four elements:

1. ACL identity: This is a string of up to 64 characters specifying the ACL name.
2. Optional **remark** entries.
3. One or more deny/permit list entries (ACEs): One entry per line.

Element	Notes
Identifier	Alphanumeric; Up to 64 Characters, Including Spaces
Remark	Allows up to 100 alphanumeric characters, including blank spaces. (If any spaces are used, the remark must be enclosed in a pair of single or double quotes.) A remark is associated with a particular ACE and will have the same sequence number as the ACE. (One remark is allowed per ACE.) Refer to "Attaching a Remark to an ACE" on page 5-53.
Maximum ACEs Per Switch	The maximum number of ACEs supported by the switch is up to 3072 for IPv6 ACEs and up to 3072 for IPv4 ACEs. The maximum number of ACEs applied to an IP routing interface, or port depends on the concurrent resource usage by multiple configured features. For more information, use the **show < qos | access-list > resources** command and/or refer to "Monitoring Shared Resources" on page 5-74.

4. Implicit Deny: Where an ACL is applied to an interface, it denies any packets that do not have a match with any of the ACEs explicitly configured in the list. The Implicit Deny does not appear in ACL configuration listings, but always functions when the switch uses an ACL to filter packets. (You cannot delete the Implicit Deny, but you can supersede it with a **permit ipv6 any any** ACE.)

Individual ACEs in an IPv6 ACL include:

- Optional remark statements

- A permit/deny statement
- Source and destination IPv6 addressing
- Choice of IPv6 criteria
- Optional ACL **log** command (for **deny** entries)

```
ipv6 access-list < identifier >

    [ seq-# ]

        [ remark < remark-str ]

        < permit | deny >
            0 - 255
            esp
            ah
            sctp
            icmp
                < SA > [operator < value >]
                < DA > [operator < value >] [type [code] | icmp-msg ]     [dscp < codepoint | precedence >]
            ipv6
            tcp
                < SA > [operator < value >]
                < DA > [operator < value >]
                    [dscp < codepoint | precedence]
                    [established]
                    [ack | fin | rst | syn]
            udp
                < SA > [operator < value >]
                < DA > [operator < value >] [dscp < codepoint | precedence ]

        [log]  (Allowed only with "deny" ACEs.)
    . . .
    < Implicit Deny Any Any >
    exit
```

Figure 5-10. General Structure Options for an IPv6 ACL

For example, the ACL in figure 5-11 filters traffic for individual hosts in some instances and all hosts in others:

```
HP Switch# show run
  .
  .
  .
ipv6 access-list "Sample-List-1"
   10 permit ipv6 2001:db8:0:130::55/128 2001:db8:0:130::240/128
   20 permit tcp ::/0 ::/0 eq 23
   30 remark "ALLOWS HTTP FROM SINGLE HOST."
   30 permit tcp 2001:db8:0:140::14/128 eq 80 ::/0 eq 3871
   40 remark "DENIES HTTP FROM ANY TO ANY."
   40 deny tcp ::/0 ::/0 eq 80 log
   50 deny udp 2001:db8:0:150::44/128 eq 69 2001:db8:0:120::19/128
              range 3680 3690 log
   60 deny udp ::/0 2001:db8:0:150::121/128 log
   70 permit ipv6 2001:db8:0:01::/56 ::/0
   exit
```

Figure 5-11. Example of a Displayed ACL Configuration

Line	Action
10	Permits all IPv6 traffic from the host at 2001:db8:0:130::55 to the host at 2001:db8:0:130::240.
20	Permits all Telnet traffic from any source to any destination.
30	Includes a remark and permits TCP port 80 traffic received at any destination as port 3871 traffic.
40	Includes a remark and denies TCP port 80 traffic received at any destination, and causes a log message to be generated when a match occurs.
50	Denies UDP port 69 (TFTP) traffic sent from the host at 2001:db8:0:150::44 to the host at 2001:db8:0:120::19 with a destination port number in the range of 3680 - 3690, and causes a log message to be generated when a match occurs.
60	Denies UDP traffic from any source to the host at 2001:db8:0:150::121, and causes a log message to be generated when a match occurs.
70	Permits all IPv6 traffic with an SA prefix of 2001:db8:0:01/56 that is not already permitted or denied by the preceding ACEs in the ACL.
Note: An implicit "Deny IPv6 any any" is automatically applied following the last line (70, in this case), and denies all IPv6 traffic not already permitted or denied by the ACEs in lines 10 through 70.	

ACL Configuration Factors

The Sequence of Entries in an ACL Is Significant

When the switch uses an ACL to determine whether to permit or deny a packet, it compares the packet to the criteria specified in the individual Access Control Entries (ACEs) in the ACL, beginning with the first ACE in the list and proceeding sequentially until a match is found. When a match is found, the switch applies the indicated action (permit or deny) to the packet. This is significant because, once a match is found for a packet, subsequent ACEs in the same ACL will not be applied to that packet, regardless of whether they match the packet.

For example, suppose that you have applied the ACL shown in figure 5-12 to inbound IPv6 traffic on VLAN 1 (the default VLAN):

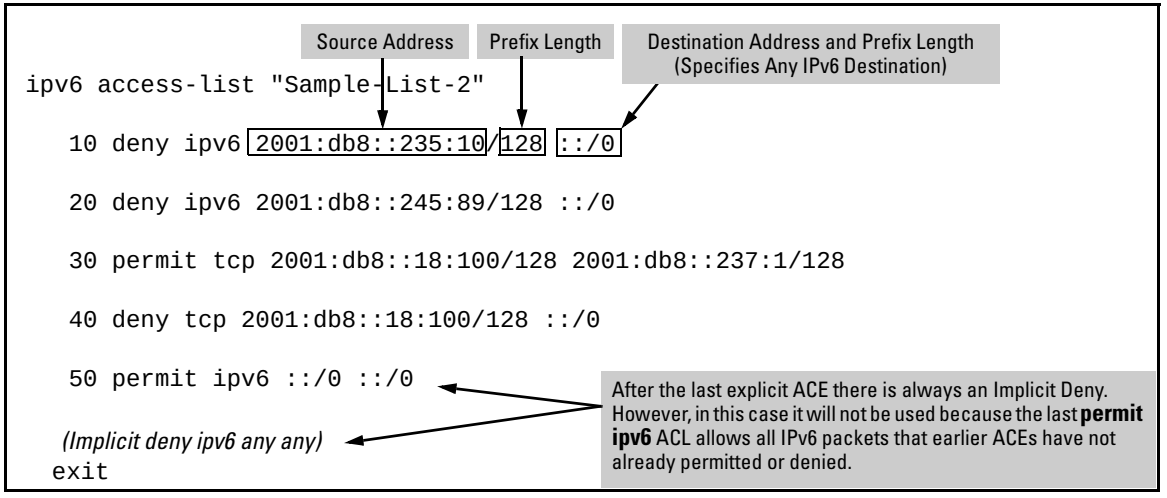


Figure 5-12. Example of an ACE that Permits All IPv6 Traffic Not Implicitly Denied

Table 5-1. Effect of the Above ACL on Inbound IPv6 Traffic in the Assigned VLAN

Line #	Action
n/a	Shows IP type (IPv6) and ID (Sample-List-2).
10	A packet from source address 2001:db8:235:10 will be denied (dropped). This ACE filters out all packets received from 2001:db8:235:10. As a result, IPv6 traffic from that device will not be allowed and packets from that device will not be compared against any later entries in the list.
20	A packet from IPv6 source address 2001:db8::245:89 will be denied (dropped). This ACE filters out all packets received from 2001:db8::245:89. As the result, IPv6 traffic from that device will not be allowed and packets from that device will not be compared against any later entries in the list.
30	A TCP packet from SA 2001:db8::18:100 with a DA of 2001:db8::237:1 will be permitted (forwarded). Since no earlier ACEs in the list have filtered TCP packets from 2001:db8::18:100 with a destination of 2001:db8::237:1, the switch will use this ACE to evaluate such packets. Any packets that meet this criteria will be forwarded. (Any packets that do not meet this TCP source-destination criteria are not affected by this ACE.)
40	A TCP packet from source address 2001:db8::18:100 to ***any*** destination address will be denied (dropped). Since, in this example, the intent is to block TCP traffic from 2001:db8::18:100 to any destination ***except*** the destination stated in the ACE at line 30, this ACE must follow the ACE at line 30. (If their relative positions were exchanged, all TCP traffic from 2001:db8::18:100 would be dropped, including the traffic for the 2001:db8::237:1 destination.)
50	Any packet from any IPv6 source address to any IPv6 destination address will be permitted (forwarded). The only traffic filtered by this ACE will be packets not specifically permitted or denied by the earlier ACEs.
n/a	The *Implicit Deny (***deny ipv6 any any***)* is a function the switch automatically adds as the last action in all IPv6 ACLs. It denies (drops) traffic from any source to any destination that has not found a match with earlier entries in the ACL. In this example, the ACE at line 50 permits (forwards) any traffic not already permitted or denied by the earlier entries in the list, so there is no traffic remaining for action by the Implicit Deny function.
exit	Defines the end of the ACL.

Allowing for the Implied Deny Function

In any ACL having one or more ACEs there will always be a packet match. This is because the switch automatically applies the Implicit Deny as the last ACE in any ACL. This function is not visible in ACL listings, but is always present. (Refer to figure 5-12.) This means that if you configure the switch to use an ACL for filtering either inbound or outbound traffic on an interface, any IPv6 packets not specifically permitted or denied by the explicit entries you create will be denied by the Implicit Deny action. If you want to preempt the Implicit Deny (so that IPv6 traffic not specifically addressed by earlier ACEs in a given ACL will be permitted), insert an explicit **permit ipv6 any any** as the last explicit ACE in the ACL.

A Configured ACL Has No Effect Until You Apply It to an Interface

The switch stores ACLs in the configuration file. Until you actually assign an ACL to an interface, it is present in the configuration, but not used (and does not use any of the monitored resources described in the appendix titled "Monitored Resources" in the latest version of the *Management and Configuration Guide* for your switch.)

You Can Assign an ACL Name to an Interface Even if the ACL Has Not Been Configured

In this case, if you subsequently create an ACL with that name, the switch automatically applies each ACE as soon as you enter it in the running-config file. Similarly, if you modify an existing ACE in an ACL you already applied to an interface, the switch automatically implements the new ACE as soon as you enter it. (See "" on page 5-74.) The switch allows up to 2048 ACLs each for IPv4 and IPv6. For example, if you configure two ACLs, but assign only one of them to a VLAN, the ACL total is two, for the two unique ACL names. If you then assign the name of an empty ACL to a VLAN, the new ACL total is three, because the switch now has three unique ACL names in its configuration.

Using the CLI To Create an ACL

Command	Page
access-list (standard ACLs)	5-40
access-list (extended ACLs)	5-40
ip access-list (named ACLs)	5-40

You can use either the switch CLI or an offline text editor to create an ACL. This section describes the CLI method, which is recommended for creating short ACLs. (To use the offline method, refer to "Creating or Editing ACLs Offline" on page 5-68.)

General ACE Rules

These rules apply to all ACEs you create or edit using the CLI:

Adding or Inserting an ACE in an ACL. To *add* an ACE to the end of an ACL, use the **ipv6 access-list < *name-str* >** command to enter the context for a specific IPv6 ACL. (If the ACL does not already exist in the switch configuration, this command creates it.) Then enter the text of the ACE without specifying a sequence number. For example, the following pair of commands enter the context of an ACL named "List-1" and add a "permit" ACE to the end of the list. This new ACE permits the IPv6 traffic from the device at 2001:db8:0:a9:8d:100 to go to all destinations.

```
HP Switch(config)# ipv6 access-list List-1
HP Switch(config-ipv6-acl)# permit host 2001:db8:0:a9::8d:100 any
```

To insert an ACE anywhere in an existing ACL, enter the context of the ACL and specify a sequence number. For example, to insert a new ACE as line 15 between lines 10 and 20 in an existing ACL named "List-2" to deny traffic from the device at 2001:db8:0:a9::8d:77, you would use the following commands:

```
HP Switch(config)# ipv6 access-list List-2
HP Switch(config-ipv6-acl)# 15 deny ipv6 host 2001:db8:0:a9::8d:77 any
```

To Delete an ACE. Enter the ACL context and delete the sequence number for the unwanted ACE. (To view the sequence numbers of the ACEs in a list, use **show access-list < *acl-name-str* > config**.) For example, to delete the ACE at line 40 in an ACL named "List-2", you would enter the following commands:

```
HP Switch(config)# ipv6 access-list List-2 config
HP Switch(config-ipv6-acl)# no 40
```

Duplicate ACE Sequence Numbers. Duplicate sequence numbering for ACEs are not allowed in the same ACL. Attempting to enter a duplicate ACE displays the **Duplicate sequence number** message.

Using CIDR Notation To Enter the IPv6 ACL Prefix Length

CIDR (Classless Inter-Domain Routing) notation is used to specify ACL prefix lengths. The switch compares the address bits specified by a prefix length for an SA or DA in an ACE with the corresponding address bits in a packet being filtered by the ACE. If the designated bits in the ACE and in the packet have identical settings, then the addresses match.

Table 5-2. Examples of CIDR Notation for Prefix Lengths

SA or DA Used In an ACL with CIDR Notation	Resulting Prefix Length Defining an Address Match	Meaning
2620:0:a03:e102::/64	2620:0:a03:e102	The leftmost 64 bits must match. The remaining 64 bits are wildcards.
2620:0:a03:e102:215::/80	2620:0:a03:e102:215	The leftmost 80 bits must match. The remaining 48 bits are wildcards.
2620:0:a03:e102:215:60ff:fe7a:adc0/128	2620:0:a03:e102:215:60ff:fe7a:adc0	All 128 bits must match. This specifies a single host address.
2001:db8:a03:e102:0:ab4:100::/112	2001:db8:a03:e102:0:ab4:100	The leftmost 112 bits must match. The remaining 16 bits are wildcards.

Configuration Commands

Command Summary for Configuring ACLs

Create an IPv6 ACL *or* Add an ACE to the End of an Existing IPv6 ACL	HP Switch(config)# ipv6 access-list < *name-str* > HP Switch(config-ipv6-acl)# < deny | permit > < ipv6 | esp | ah | sctp | *ipv6-protocol-nbr* > < any | host <*SA* > | *SA*/< *prefix-length* >> < any | host < *DA* > | *DA*/< *prefix-length* >>	5-40
	< tcp | udp > < any | host <*SA* > | *SA*/< *prefix-length* > > [*comparison-operator* < *value* >] < any | host < *DA* > | *DA*/< *prefix-length* >> [*comparison-operator* < *value* >] [established][1] [ack] [fin] [rst] [syn][2]	
	< icmp > < any | host < *SA* > | *SA* /< *prefix-length* >> < any | host < *DA* > | *DA* /< *prefix-length* >> [log]	
Insert an ACE by Assigning a Sequence Number	HP Switch(config)# ipv6 access-list < *name-str* > HP Switch(config-ipv6-acl)# < *seq-#* > < deny | permit > *The deny and permit keywords use the options shown above for "Create an IPv6 ACL".*	5-49
Delete an ACE or a Remark by Sequence Number	HP Switch(config)# ipv6 access-list < *name-str* > HP Switch(config-ipv6-acl)# no < *seq-#* > [remark] (**Note:** You can also delete an ACE by entering no < permit | deny > followed by the settings explicitly configured for that ACE.)	5-51
Resequence the ACEs in an ACL	HP Switch(config)# ipv6 access-list resequence < *name-str* > < *starting-#* > < *increment* >.	5-52

[1]TCP only.
[2]TCP flag (control bit) options for destination TCP.
[3]The **log** function is available only for "deny" ACLs, and generates a message only when there is a "deny" match.

— *Continued* —

Continued from preceding page. —

Action	Command(s)	Page
Enter a Remark	HP Switch(config)# ipv6 access-list < *name-str* > HP Switch(config-ipv6-acl)# remark < *remark-str* >	5-53 5-56
Remove a Remark: – Immediately After Entry – After entry of an ACE	 HP Switch(config-ipv6-acl)# no remark HP Switch(config-ipv6-acl)#no < *seq-#* > remark	
Delete an IPv6 ACL	HP Switch(config)# no ipv6 access-list < *name-str* > vlan	5-46
Delete an IPv6 ACL	HP Switch(config)# no ipv6 access-list < *name-str* >	5-46

Command Summary for Enabling, Disabling, and Displaying ACLs

Enable or Disable an IPv6 VACL	HP Switch(config)# [no] vlan < *vid* > ipv6 access-group < *name-str* > vlan	
Enable or Disable a Static Port ACL	HP Switch(config)# [no] interface < *port-list* | trk*x* > ipv6 access-group < *name-str* > in HP Switch(eth-< *port-list* >| trk*x* >)# [no] ipv6 access-group < *name-str* > in	
Displaying ACL Data	HP Switch(config)# show access-list HP Switch(config)# show access-list < *acl-name-str* > [config] HP Switch(config)# show access-list config HP Switch(config)# show access-list ports < *port-list* | trk*x* > HP Switch(config)# show access-list vlan < *vid* > HP Switch(config)# show access-list resources	5-59

Commands To Create, Enter, and Configure an ACL

For a match to occur with an ACE, a packet must have the source and destination IPv6 address criteria specified by the ACE.

Use the following general steps to create or add to an ACL:

1. Create and/or enter the context of a given ACL.
2. Enter the first ACE in a new ACL or append an ACE to the end of an ACL.

Syntax: ipv6 access-list < *ascii-str* >

*Places the CLI in the IPv6 ACL (***ipv6-acl***) context specified by the < ascii-str > alphanumeric identifier. This enables entry of individual ACEs in the specified ACL. If the ACL does not already exist, this command creates it.*
< ascii-str >: *Specifies an alphanumeric identifier for the ACL. Consists of an alphanumeric string of up to 64 case-sensitive characters. Including spaces in the string requires that you enclose the string in single or double quotes. For example: "***Accounting ACL***". You can also use this command to access an existing ACL. Refer to "General Editing Rules" on page 5-47*

```
HP Switch(config)# ip access-list Sample-List
HP Switch(config-ipv6-acl)#
```

Figure 5-13. Example of Entering the ACL Context

Configure ACEs in an ACL. Configuring ACEs is done after using the **ipv6 access-list <*ascii-str*>** command described to enter the IPv6 ACL (**ipv6-acl**) context of an ACL.

Syntax: (ipv6 acl context) < deny | permit > < ipv6>
< any | host < *SA* > | *SA/ prefix-length* >
< any | host < *DA* > | *DA/ prefix-length* >
[log]

Appends an ACE to the end of the list of ACEs in the current ACL. In the default configuration, ACEs are automatically assigned consecutive sequence numbers in increments of 10 and can be renumbered using **resequence** *(page 5-52).*

Note: *To insert a new ACE between two existing ACEs in an ACL, precede* **deny** *or* **permit** *with an appropriate sequence number. (Refer to "Inserting an ACE in an Existing ACL" on page 5-49.)*
For a match to occur, a packet must have the source and destination IPv6 addressing criteria specified in the ACE, as well as:

- *the protocol-specific criteria configured in the ACE, including any optional elements (described later in this section)*
- *any (optional) DSCP settings configured in the ACE*

< deny | permit >

*These keywords are used in the IPv6 (***ipv6-acl***) context to specify whether the ACE denies or permits a packet matching the criteria in the ACE, as described below.*

< any | host < *DA* > | *DA/prefix*-length >

This is the second instance of addressing in an IPv6 ACE. It follows the first (SA) instance, described earlier in this section, and defines the destination IPv6 address (DA) that a packet must carry to have a match with the ACE.

- **any** — *Allows IPv6 packets to any IPv6 DA.*
- **host** < ***DA*** > — *Specifies only packets having* ***DA*** *as the destination address. Use this criterion when you want to match only the IPv6 packets for a single DA.*
- ***DA/prefix-length*** — *Specifies packets intended for one or more contiguous subnets or contiguous addresses within a single subnet. The prefix length is in CIDR format and defines the number of leftmost bits to use in determining a match. (Refer to "Using CIDR Notation To Enter the IPv6 ACL Prefix Length" on page 5-36.) In a given ACE, the DA prefix length defines how many leftmost bits in a packet's DA must exactly match the DA configured in the ACE.*

  Example: *Refer to "Examples of Prefix-Length Applications" in the presiding syntax description.*

[log]

This option can be used after the DA to generate an Event Log message if:

- *The action is* **deny**. *(Not applicable to* **permit** *actions.)*
- *There is a match.*
- *ACL logging is enabled. (Refer to "Enabling ACL Logging on the Switch" on page 5-72.)*

For a given ACE, if **log** is used, it must be the last keyword entered.

Options for TCP and UDP Traffic in IPv6 ACLs. An ACE designed to permit or deny TCP or UDP traffic can optionally include port number criteria for either the source or destination, or both. Use of TCP criteria also allows the **established** option for controlling TCP connection traffic.

TCP: < deny | permit > tcp
< *SA* > [*comparison-operator* < *tcp-src-port* >]
< *DA* > [*comparison-operator* < *tcp-dest-port* >]
[established]
[ack] [fin] [rst] [syn]

UDP: < deny | permit > udp
< *SA* > [*comparison-operator* < *udp-src-port* >]
< *DA* > [*comparison-operator* < *udp-dest-port* >]

In an IPv6 ACL using either **tcp** *or* **udp** *as the IP packet protocol type, you can optionally apply comparison operators specifying TCP or UDP source and/or destination port numbers or ranges of numbers to further define the criteria for a match. For example:*

```
#deny tcp host fe80::119 eq 23 host fe80::155
 established
#permit tcp host 2001:db8::10.100 host
    2001:db8::15:12 eq telnet
#deny udp 2001:db8::ad5:1f4 host 2001:db8::ad0:ff3
    range 161 162
```

[*comparison-operator* < *tcp/udp-src-port* >]

To specify a TCP or UDP source port number in an ACE, (1) select a comparison operator from the following list and (2) enter the port number or a well-known port name.

Comparison Operators:

- **eq** < ***tcp/udp-port-nbr*** > — *"Equal To"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be equal to* < ***tcp/udp-port-nbr*** >.
- **gt** < ***tcp/udp-port-nbr*** > — *"Greater Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be greater than* < ***tcp/udp-port-nbr*** >.
- **lt** < ***tcp/udp-port-nbr*** > — *"Less Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be less than* < ***tcp/udp-port-nbr*** >.
- **neq** < ***tcp/udp-port-nbr*** > — *"Not Equal"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must not be equal to* ***< tcp/udp-port-nbr >***.
- **range** ***< start-port-nbr > < end-port-nbr >*** — *For a match with the ACE entry, the TCP or UDP source-port number in a packet must be in the range* <***start-port-nbr > < end-port-nbr***>.

Port Number or Well-Known Port Name:

Use the TCP or UDP port number required by your application. The switch also accepts these well-known TCP or UDP port names as an alternative to their port numbers:

- ***TCP***: bgp, dns, ftp, http, imap4, ldap, nntp, pop2, pop3, smtp, ssl, telnet
- ***UDP***: bootpc, bootps, dns, ntp, radius, radius-old, rip, snmp, snmp-trap, tftp

To list the above names, press the **[Shift] [?]** *key combination after entering an operator. For a comprehensive listing of port numbers, visit* **www.iana.org/assignments/port-numbers**.

[*comparison-operator* < *tcp-dest-port* >] [established]
[*comparison-operator* < *udp-dest-port* >]

This option, if used, is entered immediately after the **< *DA* >** *entry. To specify a TCP or UDP port number, (1) select a comparison operator and (2) enter the port number or a well-known port name.*

Comparison Operators and Well-Known Port Names — *These are the same as are used with the TCP/UDP source-port options, and are listed earlier in this command description.*

[established] *— This option applies only where TCP is the configured IPv6 protocol type. It blocks the synchronizing packet associated with establishing a new TCP connection while allowing all other IPv6 traffic for existing connections. For example, a Telnet connect requires TCP traffic to move both ways between a host and the target device. Simply applying a* **deny** *to inbound Telnet traffic on a VLAN would prevent Telnet sessions in either direction because responses to outbound requests would be blocked. However, by using the* **established** *option, inbound Telnet traffic arriving in response to outbound Telnet requests would be permitted, but inbound Telnet traffic trying to establish a new connection would be denied. The* **established** *and* **dscp** *options are mutually exclusive in a given ACE. Configuring* **established** *and any combination of TCP control bits in the same ACE is supported, but* **established** *must precede any TCP control bits configured in the ACE.*

TCP Control Bits. *In a given ACE for filtering TCP traffic you can configure one or more of these options:*

[ack] — *Acknowledgement.*

[fin] — *Sender finished.*

[rst] — *Connection reset.*

[syn] — *TCP control bit: sequence number synchronize.*

For more on using TCP control bits, refer to RFC 793.

Filtering Switched IPv6 Traffic Inbound on a VLAN

For a given VLAN interface, you can assign an ACL as a VACL to filter switched IPv6 traffic entering the switch on that VLAN.For a given VLAN interface, you can assign an ACL as a VACL to filter switched or routed IPv6 traffic entering the switch on that VLAN. You can also use the same ACL for assignment to multiple VLANs. For limits and operating rules, refer to "ACL Configuration and Operating Rules" on page 5-25.

Syntax: [no] vlan < *vid* > ipv6 access-group < *identifier* > vlan

Assigns an ACL as a VACL to a VLAN to filter switched IPv6 traffic entering the switch on that VLAN. Assigns an ACL as a VACL to a VLAN to filter switched or routed IPv6 traffic entering the switch on that VLAN. You can use either the global configuration level or the VLAN context level to assign or remove a VACL.

< *vid* >: *VLAN Identification Number.*

< *identifier* >: *The alphanumeric name by which the ACL can be accessed. An identifier can have up to 64 characters.*

The **no** *form of the command removes the ACL assignment from the interface.*

Note: *The switch allows you to assign an "empty" ACL identifier to a VLAN. In this case, if you later populate the ACL with ACEs, the new ACEs automatically become active on the assigned VLAN as they are created. Also, if you delete an assigned ACL from the switch without also using the "***no***" form of this command to remove the assignment to a VLAN, the ACL assignment remains as an "empty" ACL. For more on "empty" ACLs, refer to the notes under "Deleting an ACL" on page 5-46.*

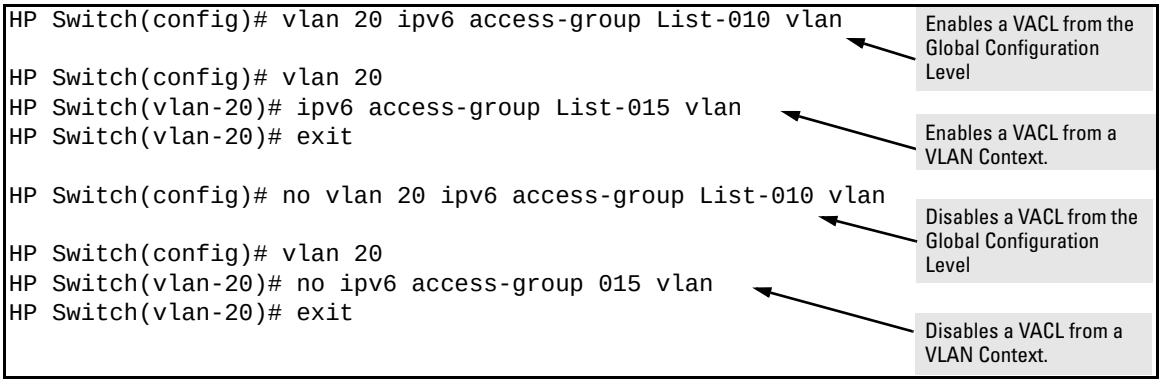


Figure 5-14. Methods for Enabling and Disabling VACLs

Deleting an ACL

Syntax: no ipv6 access-list < *identifier* >

Used in the global config context to remove the specified IPv6 ACL from the switch's running-config file.

< *identifier* >: *The alphanumeric name assigned to an ACL.*

Notes: *If an ACL name is assigned to an interface before the ACL itself has been created, then the switch creates an "empty" version of the ACL in the running configuration and assigns the empty ACL to the interface. Later adding explicit ACEs to the empty ACL causes the switch to automatically activate the ACEs as they are created and to implement the implicit deny at the end of the ACL.*

Deleting an ACL from the running configuration while the ACL is currently assigned on an interface results in an "empty" version of the ACL in the running configuration and on the interface. Later removing the ACL from the interface also removes the empty ACL from the running configuration.

Editing an Existing ACL

The CLI provides the capability for editing in the switch by using sequence numbers to insert or delete individual ACEs. An offline method is also available. This section describes using the CLI for editing ACLs. To use the offline method for editing ACLs, refer to "Creating or Editing ACLs Offline" on page 5-68.

General Editing Rules

You can use the CLI to delete individual ACEs from anywhere in an ACL, append new ACEs to the end of an ACL, and insert new ACEs anywhere within an ACL.

- When you enter a new ACE in an ACL without specifying a sequence number, the switch inserts the ACE as the last entry in the ACL.
- When you enter a new ACE in an ACL and include a sequence number, the switch inserts the ACE according to the position of the sequence number in the current list of ACEs.
- You can delete an ACE by using the **ipv6 access-list < *identifier* >** command to enter the ACL's context, and then **no < *seq-#* >** (page 5-51).
- Deleting the last ACE from an ACL leaves the ACL in the configuration as an "empty" ACL placeholder that cannot perform any filtering tasks. (In any ACL the Implicit Deny does not apply unless the ACL includes at least one explicit ACE.)

Sequence Numbering in ACLs

The ACEs in any ACL are sequentially numbered. In the default state, the sequence number of the first ACE in a list is "10" and subsequent ACEs are numbered in increments of 10. For example, the following **show run** output shows an ACL named "My-list" using the default numbering scheme:

```
ipv6 access-list "My-list"
     10 permit ipv6 2001:db8:0:5ad::25/128 ::/0
     20 permit ipv6 2001:db8:0:5ad::111/128 ::/0
     30 permit icmp 2001:db8:0:5ad::115/128 ::/0 135
     40 deny ipv6 2001:db8:0:5ad::/64 ::/0
   exit
```

Figure 5-15. Example of the Default Sequential Numbering for ACEs

An ACE can be appended to the end of the ACL by using **ipv6 access-list** from the global configuration prompt or by entering the ACL context:

```
HP Switch(config)# ipv6 access-list My-list permit esp host 2001:db8:0:5ad::19
any
```

From the global configuration prompt, appends an ACE to the end of the ACL named **My-list**..

```
 HP Switch(Config)# ipv6 access-list My-list
 HP Switch(config-ipv6-acl)# permit ipv6 any host 2001:db8:0:5ad::1
```

Enters the context of the "My-list" ACL and appends an ACE to the end of the list.

Figure 5-16. Examples of Ways to Append a New ACE to the end of an ACL

To continue from figure 5-16 and append a final ACE to the end of the ACL:

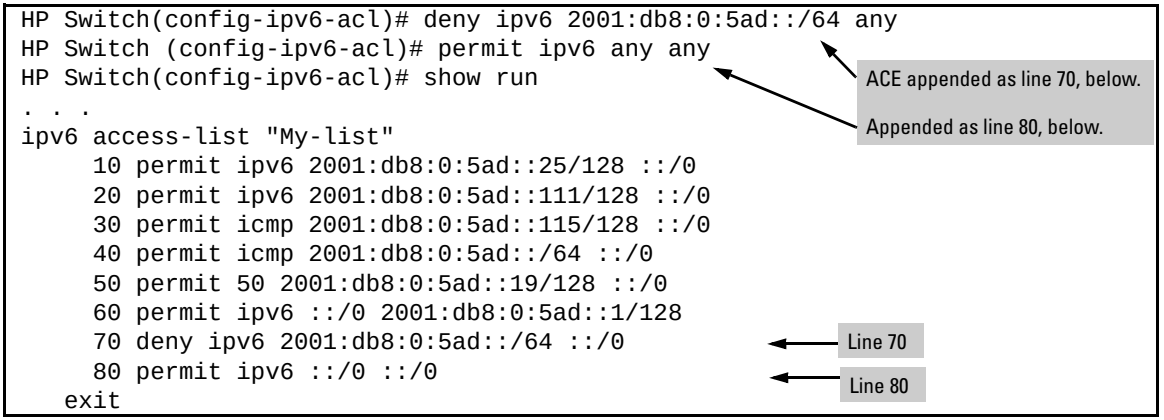


Figure 5-17. Example of Appending an ACE to an Existing List

Inserting an ACE in an Existing ACL

This action uses a sequence number to specify where to insert a new ACE into an existing sequence of ACEs in an ACL.

Syntax: <1-2147483647> < permit | deny > < *ipv6-ACE-criteria* >

Used in the context of a given ACL, this command inserts an ACE into the ACL.

<1-2147483647>: *The range of valid sequence numbers for an ACL.*

< *ipv6-ACE-criteria* >: *The various traffic selection options described earlier in this chapter.*

Note: *Entering an ACE that would result in an out-of-range sequence number is not allowed. Use the resequence command to free up ACE numbering availability in the ACL. Refer to "Resequencing the ACEs in an IPv6 ACL" on page 5-52.*

Examples of Inserting a New ACE in an Existing ACL. From the global configuration context, insert a new ACE with a sequence number of 45 between the ACEs numbered 40 and 50 in figure 5-17.

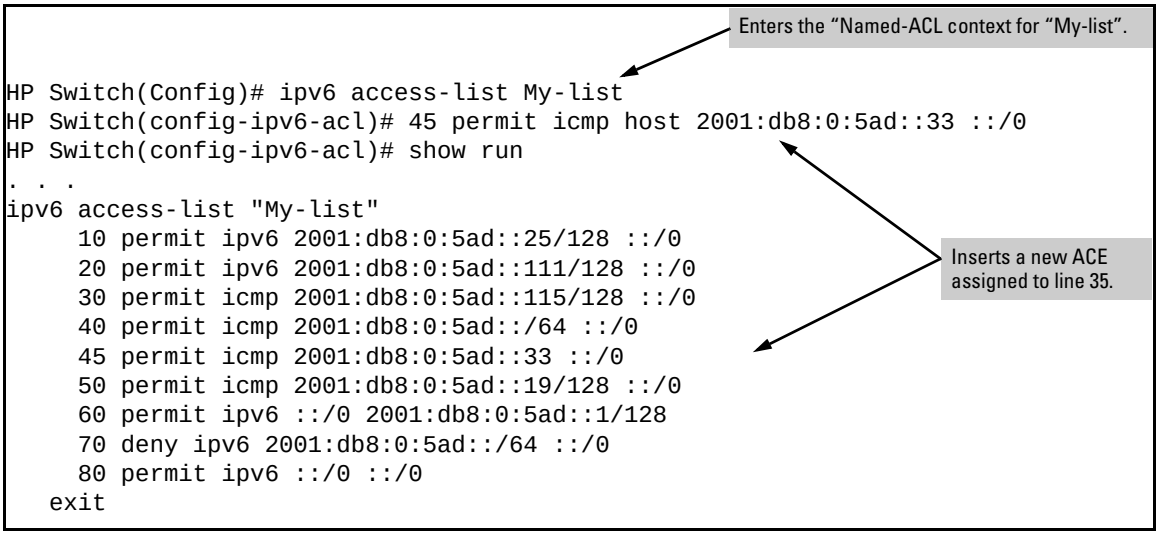


Figure 5-18. Example of Inserting an ACE in an Existing ACL

From within the context of an IPv6 ACL named "List-01", insert a new ACE between two existing ACEs. In this example, the first command creates a new ACL and enters the ACL context. The next two ACEs entered become lines 10 and 20 in the list. The third ACE entered is inserted between lines 10 and 20 by using the sequence command with a sequence number of 11.

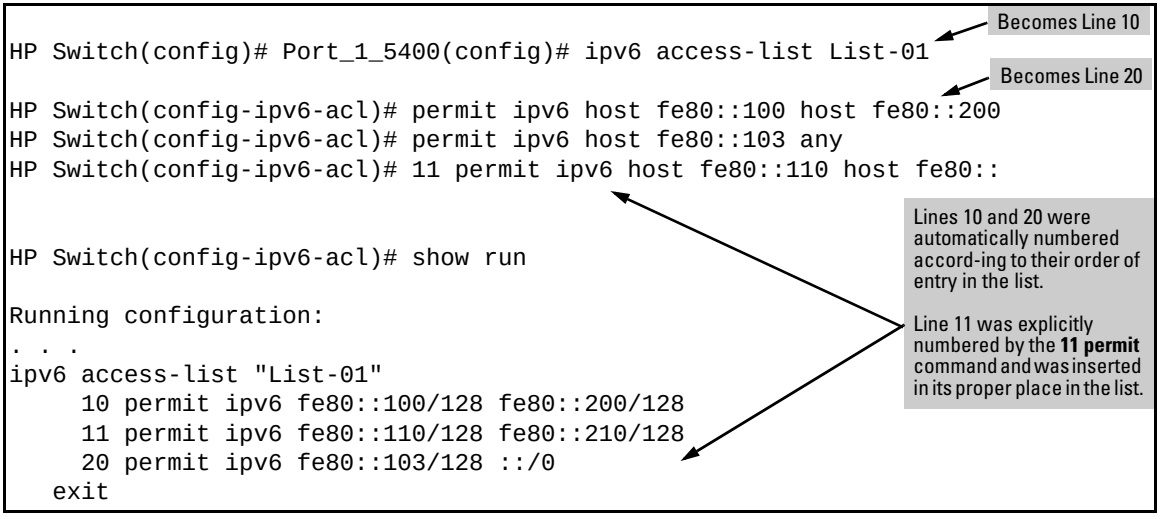


Figure 5-19. Example of Inserting an ACE into an Existing Sequence

Deleting an ACE from an Existing ACL

This action provides the option of using either the sequence number of an ACE or the syntax of the ACE to delete the ACE from an ACL.

Syntax: no <1-2147483647>

no < permit | deny > < *ipv6-ACE-criteria* >

Both command options require entering the configuration context of the ACL containing the ACE you want to delete.

The first command option deletes the ACE assigned to the specified sequence number. The second command option deletes the ACE having the syntax specified by ***< ipv6-ACE-criteria >****.*

<1-2147483647>: *The range of valid sequence numbers for an ACL.*

< ipv6-ACE-criteria >: *The traffic selection options included in the ACE. To use this method to delete an ACE, the criteria specified in the command must match the criteria specified in the actual ACE you want to delete.*

(For details on these options, refer to "Command Summary for Configuring ACLs" on page 5-38.)

1. To find the sequence number of the ACE you want to delete, use **show access-list < *identifier* >** or **show access-list config** to view the ACL.
2. Use **ipv6 access-list < *identifier* > config** to enter the IPv6 ACL (**config-ipv6-acl**) context of the specified ACE.
3. In the IPv6 ACL (**config-ipv6-acl**) context, type **no** and enter the sequence number of the ACE you want to delete.

Figure 5-20 illustrates the process for deleting an ACE from a list:

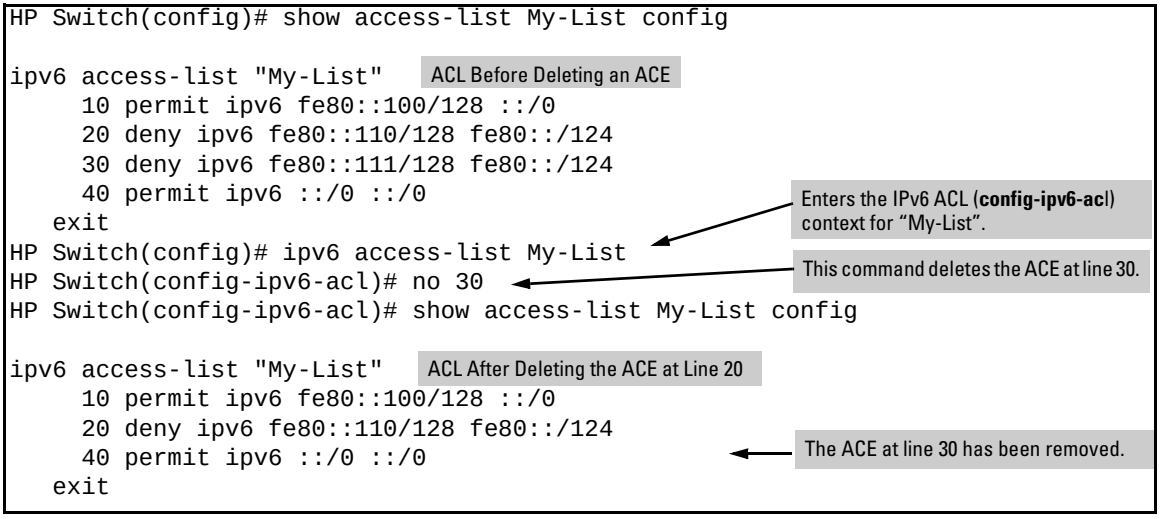


Figure 5-20. Example of Deleting an ACE from An IPv6 ACL

Resequencing the ACEs in an IPv6 ACL

This action reconfigures the starting sequence number for ACEs in an IPv6 ACL, and resets the numeric interval between sequence numbers for ACEs configured in the ACL.

Syntax: ipv6 access-list resequence < *identifier* > < *starting-seq-#* > < *interval* >

Resets the sequence numbers for all ACEs in the ACL.
< *starting-seq-#* > : *Specifies the sequence number for the first ACE in the list. (Default: 10; Range: 1 - 2147483647)*
< *interval* > : *Specifies the interval between consecutive sequence numbers for the ACEs in the list. (Default: 10; Range: 1 - 2147483647)*

1. To view the current sequence numbering in an ACE, use **show access-list config** or **show access-list < *identifier* > config**.
2. Use the command syntax (above) to change the sequence numbering.

This example resequences the "My-List" ACL at the bottom of figure 5-20 so that the list begins with line 100 and uses a sequence interval of 100.

```
HP Switch(config)# show access-list My-List config

ipv6 access-list "My-List"
     10 permit ipv6 fe80::100/128 ::/0
     20 deny ipv6 fe80::110/128 fe80::/124
     40 permit ipv6 ::/0 ::/0
   exit

HP Switch(config)# ipv6 access-list resequence My-List 100
100
HP Switch(config)# show access-list config

ipv6 access-list "My-List"
     100 permit ipv6 fe80::100/128 ::/0
     200 deny ipv6 fe80::110/128 fe80::/124
     300 permit ipv6 ::/0 ::/0
   exit
```

Figure 5-21. Example of Viewing and Resequencing an ACL

Attaching a Remark to an ACE

A remark is numbered in the same way as an ACE, and uses the same sequence number as the ACE to which it refers. This operation requires that the remark for a given ACE be entered prior to entering the ACE itself.

Syntax: remark < *remark-str* >
< 1-2147483647 > remark < *remark-str* >
no < *seq-#* > remark

These commands are used in the ACL context to enter a comment related to an adjacent ACE. To associate a remark with a specific ACE, do one of the following:

- *Enter the remark first (without a sequence number) and immediately follow it with the ACE (also without a sequence number). The remark and the following ACE will have the same (automatically generated) sequence number.*
- *Enter the ACE with or without a sequence number, then use* **<1-2147483647> remark < *remark-str* >** *to enter the remark, where a number in the range of* **<1-2147483647>** *matches the sequence number of the related ACE. This method is useful when you want to enter a remark at some time after you have entered the related ACE.*

< *remark-str* >: *The text of the remark. If spaces are included in the remark, then the remark string must be delimited by either single quotes or double quotes. For example:*

```
remark Permits_Telnet_from_2001:db8:0:1ab_subnet
remark “Permits Telnet from 2001:db8:0:1ab subnet”
remark ‘Permits Telnet from 2001:db8:0:1ab subnet’
```

<1-2147483647>: *The range of valid sequence numbers for an ACL.*

For example, if the sequence number of the last ACE entered is “30” and sequence numbering is set to the (default) interval of 10, then entering a remark and another ACE without specifying any sequence numbers results in a sequence number of “40” for both the remark and the ACE that follows it.

The **no** *form of the command deletes the indicated remark, but does not affect the related ACE.*

Appending Remarks and Related ACEs to the End of an ACL. To include a remark for an ACE that will be appended to the end of the current ACL, enter the remark first, then enter the related ACE. This results in the remark and the subsequent ACE having the same sequence number. For example, to append an ACE with an associated remark to the end of an ACL named “List-100”, you would enter remarks from the CLI context for the desired ACL:

```
HP Switch(config)# ipv6 access-list List-100
HP Switch(config-ipv6-acl)# permit tcp host 2001:db8:0:b::100:17 eq telnet any
HP Switch(config-ipv6-acl)# permit tcp host 2001:db8:0:b::100:23 eq telnet any
HP Switch(config-ipv6-acl)# remark “BLOCKS UNAUTH TELNET TRAFFIC FROM SUBNET B”
HP Switch(config-ipv6-acll)# deny tcp 2001:db8:0:a::/64 eq telnet any
HP Switch(config-ipv6-acl)# show access-list List-100 config

ipv6 access-list "List-100"
     10 remark "TEXT"
     10 permit tcp 2001:db8:0:b::100:17/128 eq 23 ::/0
     20 permit tcp 2001:db8:0:b::100:23/128 eq 23 ::/0
     30 remark "BLOCKS UNAUTH TELNET TRAFFIC FROM SUBNET B"
     30 deny tcp 2001:db8:0:b::/64 eq 23 ::/0
   exit
HP Switch(config-ipv6-acl)#
```

The remark is assigned the same number as the immediately following ACE (“30” in this example) is assigned when it is automatically appended to the end of the list. This operation applies where new remarks and ACEs are appended to the end of the ACL and are automatically assigned a sequence number.

Figure 5-22. Example of Appending a Remark and Its Related ACE to the End of an ACL

Inserting Remarks and Related ACEs Within an Existing List. To insert an ACE with a remark within an ACL by specifying a sequence number, insert the numbered remark first, then, using the same sequence number, insert the ACE. For example:

```
HP Switch(config-ipv6-acl)# 15 remark "PERMIT HTTP; STATION 23; SUBNET 1D"
HP  Switch(config-ipv6-acl)#  15  permit  tcp  host  2001:db8:0:1d::23  eq  80
2001:db8:0:2f::/64


HP Switch(config-ipv6-acl)# show access config
. . .

ipv6 access-list "List-105"
     10 permit tcp 2001:db8:0:1f::/64 eq 80 2001:db8:0:2f::/64
     15 remark "PERMIT HTTP; STATION 23; SUBNET 1D"
     15 permit tcp 2001:db8:0:1d::23/128 eq 80 2001:db8:0:2f::/64
     20 deny tcp 2001:db8:0:1d::/64 eq 80 2001:db8:0:2f::/64
   exit
. . .
```

The above two commands insert a remark with its corresponding ACE (same sequence number) between two previously configured ACEs.

Figure 5-23. Example of Inserting a Remark and an ACE Within an Existing ACL

Inserting a Remark for an ACE that Already Exists in an ACL. If an ACE already exists in a given ACL, you can insert a remark for that ACE by simply configuring the remark to have the same sequence number as the ACE.

Replacing an Existing Remark. To replace an existing remark in a given ACL:

1. Use **ipv6 access-list < *identifier* >** to enter the desired ACL context.
2. Configure the replacement remark with the same sequence number as the remark you want to replace. This step overwrites the former remark text with the new remark text.

For example, to change the text of the remark at line 15 in figure 5-23 to "PERMIT HTTP FROM ONE STATION", you would use the following command:

```
HP Switch(config): ipv6 access-list List-105
HP Switch(config-ipv6-acl): 15 remark "PERMIT HTTP FROM ONE STATION"
```

Removing a Remark from an Existing ACE. If you want to remove a remark, but want to retain the ACE, do the following:

1. Use **ipv6 access-list < *identifier* >** to enter the desired ACL context.
2. Use **no <1-2147483647> remark**.

Using the **no <1-2147483647>** command without the remark keyword deletes both the remark and the ACE to which it is attached.

Operating Notes for Remarks

- An "orphan" remark is a remark that does not have an ACE counterpart with the same sequence number. The **resequence** command renumbers an orphan remark as a sequential, standalone entry without a permit or deny ACE counterpart.

```
ipv6 access-list "XYZ"
     10 remark "Permits HTTP"
     10 permit tcp 2001:db8::2:1/120 eq 80 ::/0
     12 remark "Denies HTTP from subnet 1."
     18 remark "Denies pop3 from 1:157."
     18 deny tcp 2001:db8::1:157/128 eq 110 ::/0 log
     50 permit ipv6 ::/0 ::/0
  exit
HP Switch# ipv6 access-list resequence XYZ 100 10
HP Switch# show access-list XYZ config
ipv6 access-list "XYZ"
     100 remark "Permits HTTP"
     100 permit tcp 2001:db8::2:1/120 eq 80 ::/0
     110 remark "Denies HTTP from subnet 1."
     120 remark "Denies pop3 from 1:157."
     120 deny tcp 2001:db8::1:157/128 eq 110 ::/0 log
     130 permit ipv6 ::/0 ::/0
  exit
```

- Entering either an unnumbered remark followed by a manually numbered ACE (using **<1-2147483647>**), or the reverse (an unnumbered ACE followed by a manually numbered remark) can result in an "orphan" remark.
- Configuring two remarks without including either sequence numbers or an intervening, unnumbered ACE results in the second remark overwriting the first.

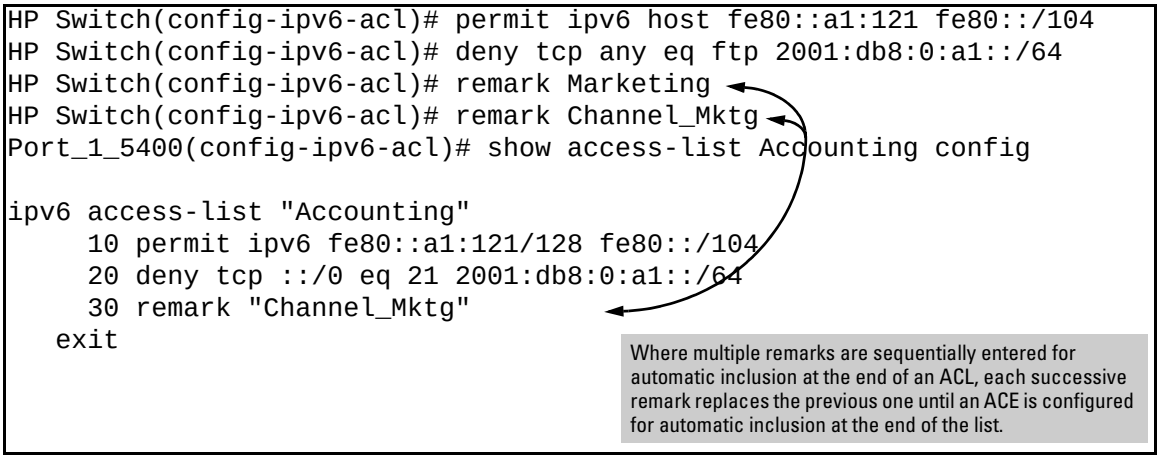


Figure 5-24. Example of Overwriting One Remark with Another

Displaying ACL Configuration Data

ACL Commands	Function	Page
show access-list	View a brief listing of all ACLs on the switch.	5-59
show access-list config	Display the ACL lists configured in the switch.	5-60
show access-list vlan < *vid* >	List the name and type for each IPv4 and IPv6 ACL application assigned to a particular VLAN on the switch.	5-61
show access-list ports < all | < *interface* >>	List the name and type of ACLs assigned to all ports on the switch or to a particular port or static trunk configured on the switch.	5-61
show access-list < *acl-name-string* >	Display detailed content information for a specific ACL.	5-63
show config show running	**show config** includes configured ACLs and assignments existing in the startup-config file. **show running** includes configured ACLs and assignments existing in the running-config file.	

Display an ACL Summary

This command lists the configured IPv4 and IPv6 ACLs, regardless of whether they are assigned to any interfaces.

Syntax: show access-list

List a summary table of the name, type, and application status of all ACLs (IPv4 and IPv6) configured on the switch.

For example:

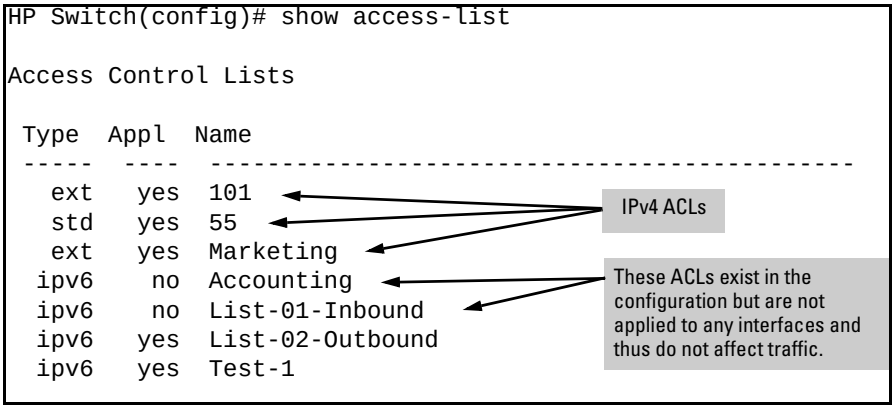


Figure 5-25. Example of a Summary Table of Access Lists

Term	Meaning
Type	Shows whether the listed ACL is an IPv6 (**ipv6**) ACL or one of two IPv4 ACL types: **std** (Standard; source-address only) or **ext** (Extended; source, and destination data).
Appl	Shows whether the listed ACL has been applied to an interface (**yes**/**no**).
Name	Shows the identifier assigned to each ACL configured in the switch.

Display the Content of All ACLs on the Switch

This command lists the configuration details for every ACL configured in the running-config file, regardless of whether you have assigned any to filter traffic on switch interfaces.

Syntax: show access-list config

List the configured syntax for all ACLs currently configured on the switch.

Note

Notice that you can use the output from this command for input to an offline text file in which you can edit, add, or delete ACL commands. Refer to "Creating or Editing ACLs Offline" on page 5-68.

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, with two ACLs configured in the switch, you will see results similar to the following:

```
HP Switch(config)# show access-list config

ip access-list extended "101"
   10 permit tcp 10.30.133.27 0.0.0.0 0.0.0.0 255.255.255.255
   20 permit tcp 10.30.155.101 0.0.0.0 0.0.0.0 255.255.255.255
   30 deny ip 10.30.133.1 0.0.0.0 0.0.0.0 255.255.255.255 log
   40 deny ip 10.30.155.1 0.0.0.255 0.0.0.0 255.255.255.255
   exit
ipv6 access-list "Accounting"
     10 permit tcp 2001:db8:0:1af::10:14/128 ::/0 eq 23
     20 permit tcp 2001:db8:0:1af::10:23/128 ::/0 eq 23
     30 deny tcp 2001:db8:0:1af::10/116 ::/0 log
     40 permit ipv6 2001:db8:0:1af::10/116 ::/0
     50 deny ipv6 ::/0 ::/0 log
   exit
```

Figure 5-26. Example of an ACL Configured Syntax Listing

Display the ACL Assignments for an Interface

This command briefly lists the identification and type(s) of ACLs currently assigned to a particular interface (one or more ports and/or trunks) in the running-config file. (The switch allows up to one, inbound ACL assignment per interface.)

Syntax: show access-list ports < *interface* >

List the ACLs assigned to interfaces in the running config file.

Note

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, the following output shows that inbound, routed IPv6 traffic and outbound, routed IPv4 traffic are both filtered on VLAN 20.

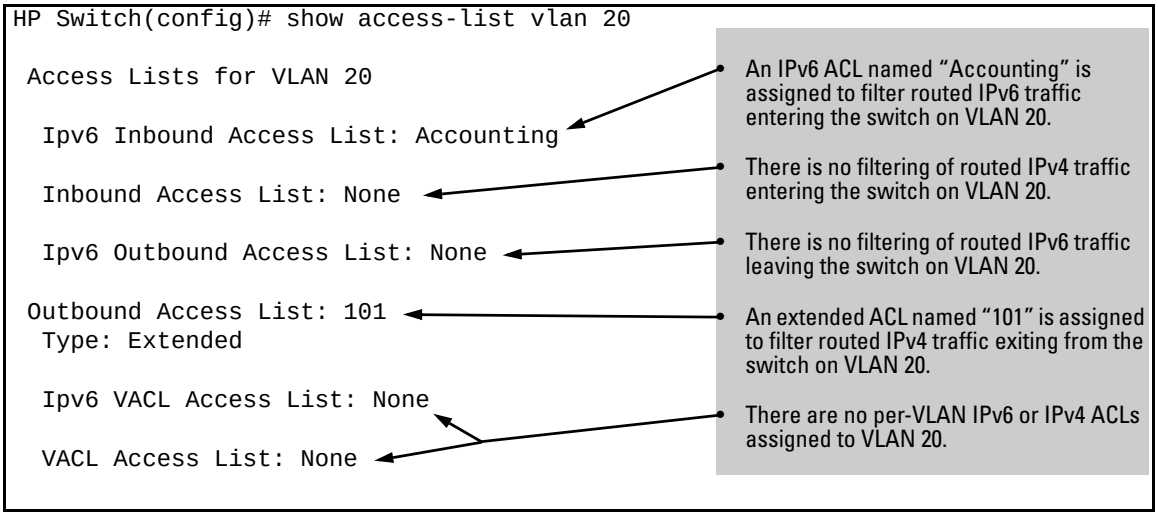


Figure 5-27. Example of Listing the ACL Assignments for a VLAN

Display Static Port (and Trunk) ACL Assignments

This command lists the identification and type(s) of current static port ACL assignments to individual switch ports and trunks, as configured in the running-config file. (The switch allows one static port ACL assignment per port.)

Syntax: show access-list ports < all | *port-list* >

Lists the current static port ACL assignments for ports and trunks in the running config file.

Note

This information also appears in the **show running** output. If you execute **write memory** after configuring an ACL, it also appears in the **show config** output.

For example, the following output shows IPv4 and IPv6 ACLs configured on various ports and trunks on the switch:

```
HP Switch(config)# show access-list ports all

Access Lists for Port B1

  Inbound Ipv6: List-01-Inbound

Access Lists for Port B12

  Inbound  : 101
  Type     : Extended
  Inbound Ipv6: Accounting

Access Lists for Port Trk2

  Inbound Ipv6: Accounting

Access Lists for Port Trk5

  Inbound  : Marketing
  Type     : Extended
```

An IPv6 ACL is filtering inbound traffic on port B1.

Both an IPv4 ACL and an IPv6 ACL are filtering inbound IPv4 and IPv6 traffic, respectively, on port B12.

An IPv6 ACL is filtering inbound IPv6 traffic on Trunk 2 (Trk2).

An IPv4 ACL is filtering inbound IPv4 traffic on Trunk 5 (Trk5).

Figure 5-28. xample of Listing the ACL Assignments for Ports and Trunks

Displaying the Content of a Specific ACL

This command displays a specific IPv6 or IPv4 ACL configured in the running config file in an easy-to-read tabular format.

Note

This information also appears in the **show running** display. If you execute **write memory** after configuring an ACL, it also appears in the **show config** display.

For information on IPv4 ACL operation, refer to the latest version of the *Access Security Guide* for your switch.

Syntax: show access-list < *identifier* > [config]

Display detailed information on the content of a specific ACL configured in the running-config file.

For example, suppose you configured the following two ACLs in the switch:

Identifier	Type	Desired Action
Accounting	IPv6	• Permit Telnet traffic from these two IPv6 addresses: – 2001:db8:0:1af::10: 14 – 2001:db8:0:1af::10: 24 • Deny Telnet traffic from all other devices in the same subnet. • Permit all other IPv6 traffic from the subnet. • Deny and log any IPv6 traffic from any other source.
List-120	IPv4 Extended	• Permit any TCP traffic from 10.30.133.27 to any destination. • Deny any other IP traffic from 10.30.133.(1-255). • Permit all other IP traffic from any source to any destination.

Use **show access-list < *identifier* >** to inspect a specific IPv6 or IPv4 ACL, as follows:

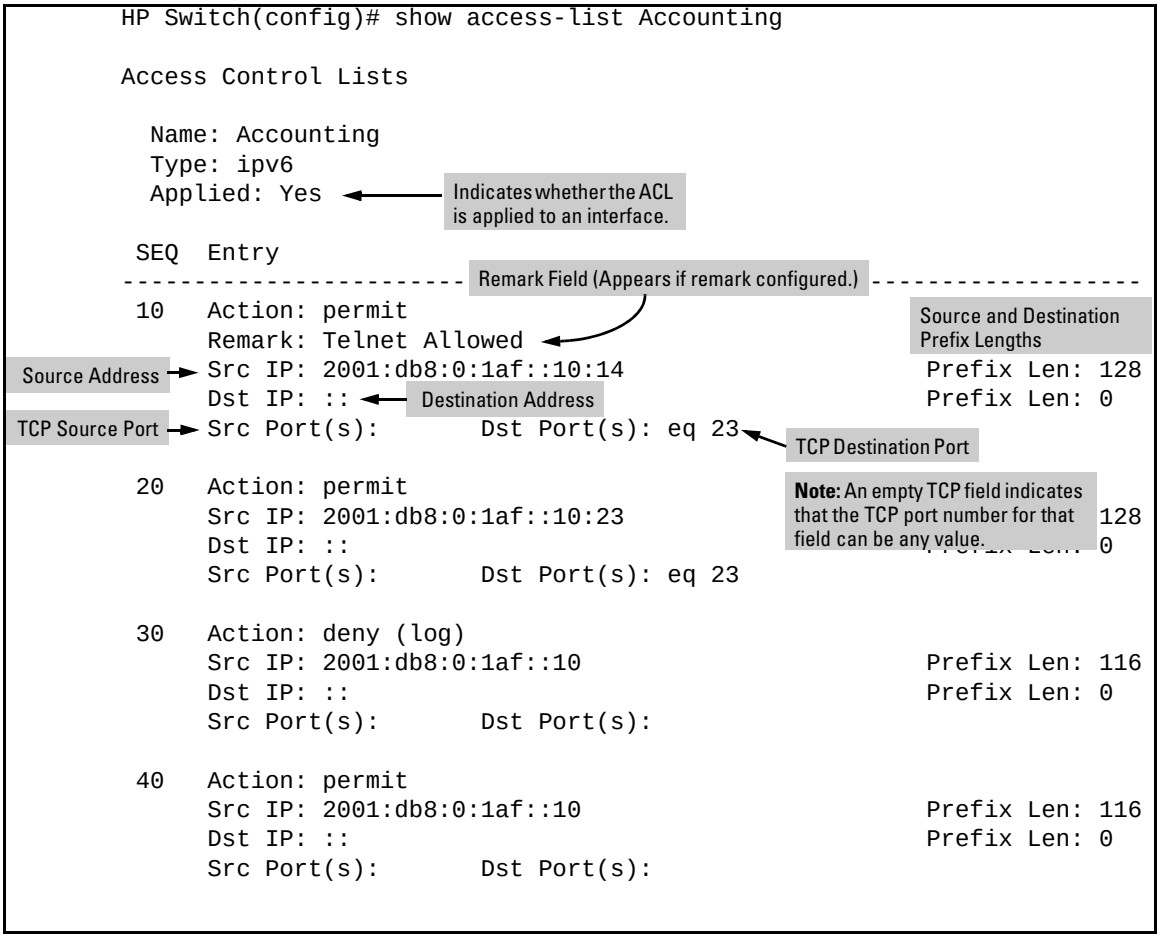


Figure 5-29. Example of Listing an IPv6 ACL

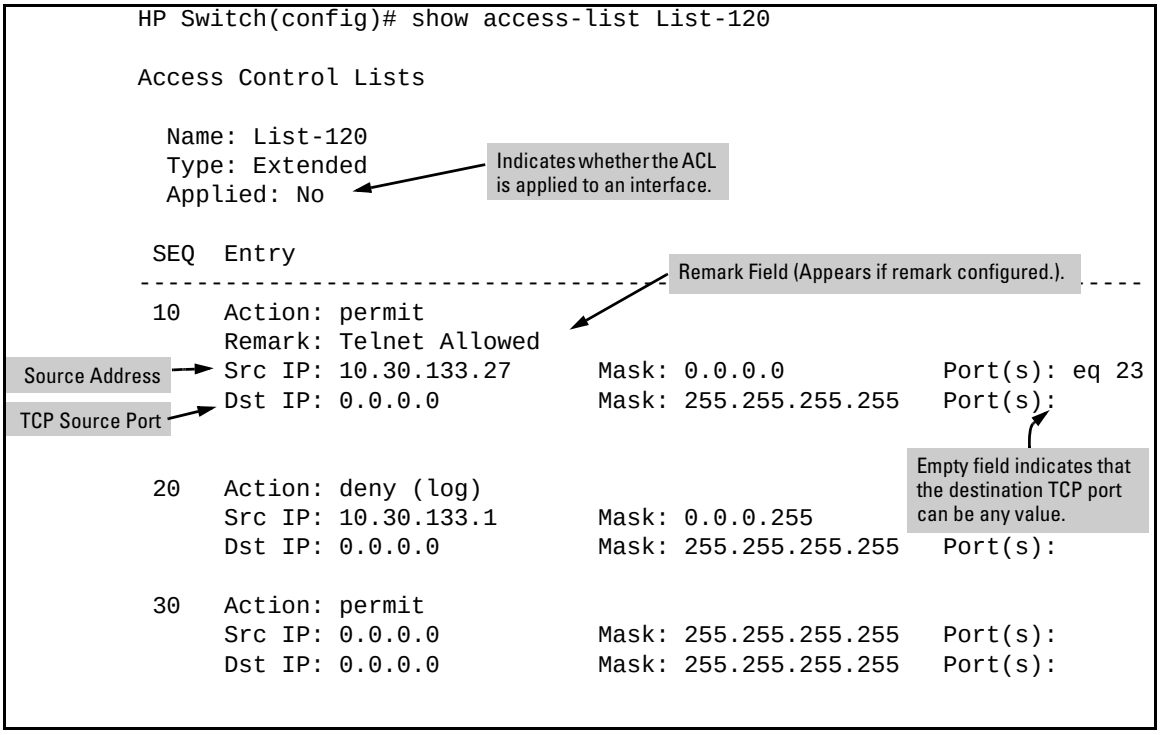


Figure 5-30. Example of Listing an IPv4 Extended ACL

The **show access-list < *identifier* > config** command shows the same ACL data as **show access-list < *identifier* >** but in the format used by the **show < run | config >** commands to list the switch configuration. For example:

```
HP Switch(config)# show access-list List-120 config

ip access-list extended "List-120"
   10 remark "Telnet Allowed"
   10 permit tcp 10.30.133.27 0.0.0.0 eq 23 0.0.0.0 255.255.255.255 precedence 0
 established
   20 deny ip 10.30.133.1 0.0.0.255 0.0.0.0 255.255.255.255 log
   30 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
```

Figure 5-31. Example of an ACL Listed with the "Config" Option

Table 5-3. Descriptions of Data Types Included in Show Access-List < *acl-id* > Output

Field	Description
Name	The ACL identifier. For IPv6 ACLs, is an alphanumeric name. For IPv4 ACLs, can be a number from 1 to 199, or an alphanumeric name.
Type	IPv6, Standard, or Extended. IPv6 ACLs use a source and a destination address, plus IPv6 protocol specifiers. Standard ACLs are IPv4 only, and use only a source IP address. Extended ACLs are available in IPv4 only, and use both source and destination IP addressing, as well as other IP protocol specifiers.
Applied	"Yes" means the ACL has been applied to an interface. "No" means the ACL exists in the switch configuration, but has not been applied to any interface, and is therefore not in use.
SEQ	The sequential number of the Access Control Entry (ACE) in the specified ACL.
Entry	Lists the content of the ACEs in the selected ACL.
Action	Permit (forward) or deny (drop) a packet when it is compared to the criteria in the applicable ACE and found to match. Includes the optional **log** option, if used, in **deny** actions.
Remark	Displays any optional remark text configured for the selected ACE.
IP	**Used for IPv4 Standard ACEs:** The source IPv4 address to which the configured mask is applied to determine whether there is a match with a packet.
Src IP	**Used for IPv6 ACEs and IPv4 Extended ACEs:** The source IPv6 or IPv4 address to which the configured mask is applied to determine whether there is a match with a packet.
Dst IP	**Used for IPv6 ACEs and IPv4 Extended ACEs:** The source and destination IP addresses to which the corresponding configured masks are applied to determine whether there is a match with a packet.
Mask	Used in IPv4 ACEs, the mask is configured in an ACE and applied to the corresponding IP address in the ACE to determine whether a packet matches the filtering criteria.
Prefix Len (source and destination)	Used in IPv6 ACEs to specify the number of consecutive high-order (leftmost) bits of the source and destination addresses configured in an ACE to be used to determine a match with a packet being filtered by the ACE.
Port(s)	Used in IPv4 extended ACEs to show any TCP or UDP operator and port number(s) included in the ACE.
Src Port(s) Dst Port(s)	Used in IPv6 ACEs to show TCP or UDP source and destination operator and port number(s) included in the ACE.

Display All ACLs and Their Assignments in the Switch Startup-Config File and Running-Config File

The **show config** and **show running** commands include in their listings any configured ACLs and any ACL assignments to interfaces. Remember that **show config** lists the startup-config file and **show running** lists the running-config file.

Creating or Editing ACLs Offline

The section titled “Editing an Existing ACL” on page 5-47 describes how to use the CLI to edit an ACL, and is most applicable in cases where the ACL is short or there is only a minor editing task to perform. The offline method provides a useful alternative to using the CLI for creating or extensively editing a large ACL.This section describes how to:

- move an existing ACL to a TFTP server
- use a text (.txt) file format to create a new ACL or edit an existing ACL offline
- use TFTP to load an offline ACL into the switch’s running-config

For longer ACLs that may be difficult or time-consuming to accurately create or edit in the CLI, you can use the offline method described in this section.

Note

The **copy** commands that used either **tftp** or **xmodem**, also include an option to use **usb** as a source or destination device for file transfers. So although the following example highlights tftp, remember that **xmodem** or **usb** can also be used to transfer ACLs to and from the switch.

Creating or Editing an ACL Offline

The Offline Process

1. Begin by doing one of the following:
   - To edit one or more existing ACLs, use **copy command-output tftp** to copy the current version of the ACL configuration to a file in your TFTP server. For example, to copy the ACL configuration to a file named **acl-001.txt** in the TFTP directory on a server at FE80::2a1:200.

     ```
     HP Switch# copy command-output 'show access-list
     config' tftp fe80::2a1:200 acl-001.txt pc
     ```
   - To create a new ACL, open a text (.txt) file in the appropriate directory on a TFTP server accessible to the switch.
2. Use a text editor to create or edit the ACL(s) in the ***.txt** ASCII file format.

If you are replacing an ACL on the switch with a new ACL that uses the same number or name syntax, begin the command file with a **no ip access-list** command to remove the earlier version of the ACL from the switch's running-config file. Otherwise, the switch will append the new ACEs in the ACL you download to the existing ACL. For example, if you planned to use the **copy** command to *replace* an ACL named "List-120", you would place this command at the beginning of the edited file:

```
no ipv6 access-list List-120
```

```
no ipv6 access-list List-120
ip access-list "List-120"
   10 remark "THIS ACE ALLOWS TELNET"
   10 permit tcp fe80::17/128 ::/0 eq 23
   20 deny ipv6 fe80::123/128 fe80::/125 log
   30 deny ipv6 fe80::255/128 fe80::/125 log
   40 remark "THIS IS THE FINAL ACE IN THE LIST"
   40 permit ipv6 ::/0 ::/0
  exit
```

Removes an existing ACL and replaces it with a new version with the same identifier. To append new ACEs to an existing ACL instead of replacing it, you would omit the first line and ensure that the sequence numbering for the new ACEs begin with a number greater than the highest number in the existing list.

Figure 5-32. Example of an Offline ACL File Designed To Replace An Existing ACL

3. Use **copy tftp command-file** to download the file as a list of commands to the switch.

```
HP Switch(config)# copy tftp command-file fe80::1ad:17 acl-001.txt pc
Running configuration may change, do you want to continue [y/n]?  y
  1. ipv6 access-list "acl-001"
  6.    ; CREATED ON JUNE 10
 10.    10 remark "Telnet Denied Here"
 13.    10 deny tcp 2001:db8:0:1af::/64 ::/0 eq 23
 16.    30 deny tcp ::/0 ::/0 log
 19.    40 deny icmp 2001:db8:0:1af::/64 ::/0 134
 22.    50 deny icmp 2001:db8:0:1af::/64 ::/0 133
 27.    ; PERMITS IPV6 ANY ANY
 31.    60 permit ipv6 ::/0 ::/0
 34.    exit
 36.    vlan 20 ipv6 access-group acl-001 vlan
```

Note: Blank lines may appear in the command output when you copy the command file to the switch. However, they are eliminated in the copy of the ACL in switch memory. This is normal operation. (See also figure 5-34 for the configuration resulting from this output.)

Figure 5-33. Example of Using "copy tftp command-file" To Configure an ACL in the Switch

4. In this example, the command to assign the ACL to a VLAN was included in the .txt command file. If this is not done in your applications, then the next step is to manually assign the new ACL to the intended VLAN.

   vlan < *vid* > ipv6 access-group < *identifier* > vlan

   vlan < *vid* > ipv6 access-group < *identifier* > in

5. You can then use the **show run** or **show access-list config** command to inspect the switch configuration to ensure that the ACL was properly downloaded.

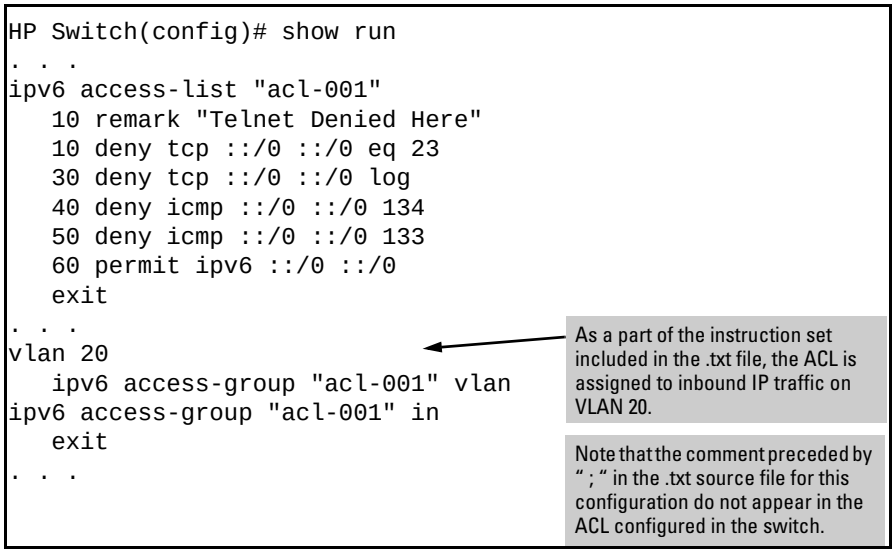


Figure 5-34. Example of Verifying the .txt File Download to the Switch

6. If the configuration appears satisfactory, save it to the startup-config file:

   ```
   HP Switch(config)# write memory
   ```

Enable IPv6 ACL "Deny" Logging

ACL logging enables the switch to generate a message when IP traffic meets the criteria for a match with an ACE that results in an explicit "deny" action. You can use ACL logging to help:

- Test your network to help ensure that your ACL configuration is detecting and denying the incoming IPv6 traffic you do not want to enter the switch.
- Receive notification when the switch denies inbound IPv6 traffic you have designed your ACLs to reject (deny).

The switch sends ACL messages to Syslog and optionally to the current console, Telnet, or SSH session. You can use **logging < >** to configure up to six Syslog server destinations.

Requirements for Using IPv6 ACL Logging

- The switch configuration must include an ACL (1) assigned to a port, trunk, or static VLAN interface and (2) containing an ACE configured with the **deny** action and the **log** option.
- For IPv6 ACL logging to a Syslog server:
  - The server must be accessible to the switch and identified in the running configuration.
  - The logging facility must be enabled for Syslog.
  - Debug must be configured to:
    - support ACL messages
    - send debug messages to the desired debug destination

  These requirements are described in more detail under "Enabling ACL Logging on the Switch" on page 5-72.

ACL Logging Operation

When the switch detects a packet match with an ACE and the ACE includes both the **deny** action and the optional **log** parameter, an ACL log message is sent to the designated debug destination. The first time a packet matches an ACE with **deny** and **log** configured, the message is sent immediately to the destination and the switch starts a wait-period of approximately five minutes.

(The exact duration of the period depends on how the packets are internally routed.) At the end of the collection period, the switch sends a single-line summary of any additional "deny" matches for that ACE (and any other "deny" ACEs for which the switch detected a match). If no further log messages are generated in the wait-period, the switch suspends the timer and resets itself to send a message as soon as a new "deny" match occurs. The data in the message includes the information illustrated in figure 5-35.

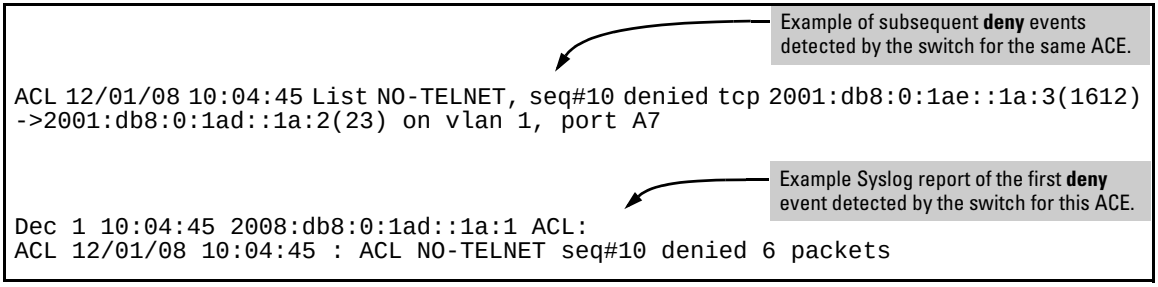


Figure 5-35. Content of a Message Generated by an ACL-Deny Action

Enabling ACL Logging on the Switch

1. If you are using a Syslog server, use the **logging < *ip-addr* >** command to configure the Syslog server IP address(es). Ensure that the switch can access any Syslog server(s) you specify.
2. Use **logging facility syslog** to enable the logging for Syslog operation.
3. Use the **debug destination** command to configure one or more log destinations. (Destination options include **logging** and **session**. For more information on debug, refer to "Debug and Syslog Messaging Operation" in appendix C, "Troubleshooting", in the latest *Management and Configuration Guide* for your switch.)
4. Use **debug acl** or **debug all** to configure the debug operation to include ACL messages.
5. Configure an ACL with the **deny** action and the **log** option in one or more ACEs.

For example, suppose that you want to do the following:

- On port 10, configure an extended ACL with an ACL-ID of 143 to deny Telnet traffic from IP address 10.38.100.127.
- Configure the switch to send an ACL log message to the console and to a Syslog server at IP address 10.38.110.54 on port 11 if the switch detects a match denying Telnet access from 10.38.100.127.

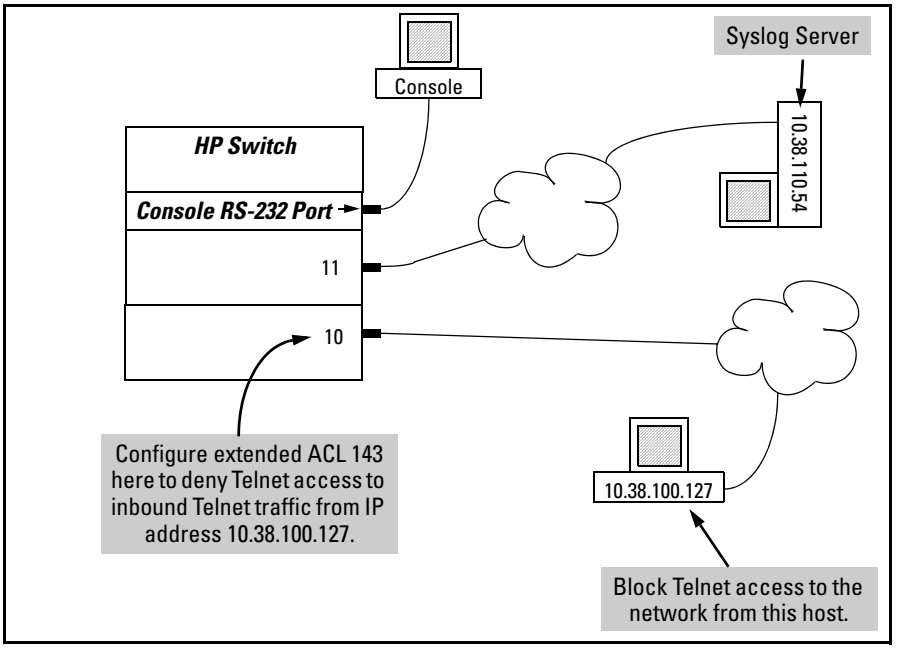


Figure 5-36. Example of an ACL Log Application

```
HP Switch(config)# access-list 143 deny tcp host 10.38.100.127 any eq telnet
log
HP Switch(config)# access-list 143 permit ip any any
HP Switch(config)# interface 10 access-group 143 in
HP Switch(config)# logging 10.38.110.54
HP Switch(config)# debug ac1
HP Switch(config)# debug destination logging
HP Switch(config)# debug destination session
HP Switch(config)# write memory

HP Switch(config)# show debug
Debug Logging
 Destination:
  Logging
   10.38.110.54
  Session
 Enabled debug types:
  event
  acl log
```

Figure 5-37. Commands for Applying an ACL with Logging

General ACL Operating Notes

ACLs do not provide DNS hostname support. ACLs cannot be configured to screen hostname IP traffic between the switch and a DNS.

ACLs Do Not Affect Serial Port Access. ACLs do not apply to the switch's serial port.

AACL Logging.

- The ACL logging feature generates a message only when packets are explicitly denied as the result of a match, and not when explicitly permitted or implicitly denied. To help test ACL logging, configure the last entry in an ACL as an explicit **deny** statement with a **log** statement included, and apply the ACL to an appropriate port or IP routing interface.
- Logging enables you to selectively test specific devices or groups. However, excessive logging can affect switch performance. For this reason, HP recommends that you remove the logging option from ACEs for which you do not have a present need. Also, avoid configuring logging where it does not serve an immediate purpose. (Note that ACL logging is not designed to function as an accounting method.) See also "Apparent Failure To Log All 'Deny' Matches" in the section titled "ACL Problems", found in appendix C, "Troubleshooting" of the latest *Management and Configuration Guide* for your switch.
- When configuring logging, you can reduce excessive resource use by configuring the appropriate ACEs to match with specific hosts instead of entire subnets. (For more on resource usage, refer to "Monitoring Shared Resources" on page 5-74.)

Minimum Number of ACEs in an IPv6 ACL. An IPv6 ACL must include at least one ACE to enable traffic screening. An IPv6 ACL can be created "empty"; that is, without any ACEs. However if an empty ACL applied to an interface, the Implicit Deny function does not operate, and the ACL has no effect on traffic.

Monitoring Shared Resources. Applied ACLs share internal switch resources with several other features. However, if the internal resources become fully subscribed, additional ACLs cannot be applied until the necessary resources are released from other applications. For information on

determining current resource availability and usage, refer to appendix E, "Monitoring Resources" in the latest *Management and Configuration Guide* for your switch. See also the appendix titled "Scalability and System Maximums" in the same guide.

Replacing or Adding To an Active IPv6 ACL Policy. If you assign an IPv6 ACL to an interface and subsequently add or replace ACEs in that ACL, each new ACE becomes active when you enter it. If the ACL is configured on multiple interfaces when the change occurs, then the switch resources must accommodate all applications of the ACL. If there are insufficient resources to accommodate one of several ACL applications affected by the change, then the change is not applied to any of the interfaces and the previous version of the ACL remains in effect.

"Strict" IPv6 TCP and UDP. When the IPv6 ACL configuration includes TCP or UDP options, the switch operates in "strict" TCP and UDP mode for increased control. In this case, the switch compares all IPv6 TCP and UDP packets against the IPv6 ACLs.

Connection-Rate ACLs. This ACL connection-rate ACLs are supported for IPv4 ACLs, but not for IPv6 ACLs.

Unable to Delete an Empty ACL in the Running Configuration. The **no vlan < *vid* > ipv6 access-group < *name-str* > vlan** command does not delete the named ACL if the ACL is currently assigned to an interface.

Unable to Delete an ACL in the Running Configuration

Attempting to delete an ACL that is currently assigned to an interface removes all configured ACEs from the ACL, but leaves an "empty" ACL in the configuration. To delete an ACL that is currently assigned to an interface, do the following:

In the interface context, use the no ipv6 access-group command to remove the ACL from the interface.

Use the **no ipv6 access-list < *name-str* >** command to delete the ACL.

IPv6 Diagnostic and Troubleshooting

Introduction

Feature	Default	CLI
IPv6 ICMP Message Interval and Token Bucket	100 ms 10 max tokens	6-2
ping6	Enabled	
traceroute6	n/a	

The IPv6 ICMP feature enables control over the error and informational message rate for IPv6 traffic, which can help mitigate the effects of a Denial-of-service attack. Ping6 enables verification of access to a specific IPv6 device, and traceroute6 enables tracing the route to an IPv6-enabled device on the network.

ICMP Rate-Limiting

ICMP rate-limiting controls the rate at which ICMPv6 generates error and informational messages for features such as:

- neighbor solicitations
- neighbor advertisements
- multicast listener discovery (MLD)
- path MTU discovery (PMTU)
- duplicate address discovery (DAD)
- neighbor unreachability detection (NUD)
- router discovery
- neighbor discovery (NDP)

ICMPv6 error message generation is enabled by default. The rate of message generation can be adjusted, or message generation can be disabled.

Controlling the frequency of ICMPv6 error messages can help to prevent DoS (Denial- of- Service) attacks. With IPv6 enabled on the switch, you can control the allowable frequency of these messages with ICMPv6 rate-limiting.

Syntax:. ipv6 icmp error-interval < 0 - 2147483647 > [bucket-size < 1 - 200 >]
no ipv6 icmp error-interval

This command is executed from the global configuration level, and uses a "token bucket" method for limiting the rate of ICMP error and informational messages. Using this method, each ICMP message uses one token, and a message can be sent only if there is a token available. In the default configuration, a new token can be added every 100 milliseconds, and a maximum of 10 tokens are allowed in the token bucket. If the token bucket is full, a new token cannot be added until an existing token is used to enable sending an ICMP message. You can increase or decrease both the frequency with which used tokens can be replaced and (optionally) the number of tokens allowed to exist.

error-interval: *Specifies the time interval in milliseconds between successive token adds. Increasing this value decreases the rate at which tokens can be added. A setting of* **0** *disables ICMP messaging.*
Default: *100;* ***Range:*** *0 - 2147483647.*

bucket-size: *This optional keyword specifies the maximum number of tokens allowed in the token bucket at any time. Decreasing this value decreases the maximum number of tokens that may be available at any time.*
Default: *10;* ***Range:*** *1 - 200.*

You can change the rate at which ICMP messages are allowed by changing the error-interval with or without a corresponding change in the bucket-size.

The **no ipv6 icmp error-interval** *command resets both the* **error-interval** *and the* **bucket-size** *values to their defaults.*

Use the **show run** *command to view the current ICMP error interval settings.*

For example, the following command limits ICMP error and informational messages to no more than 20 every 1 second:

```
HP Switch(config)# ipv6 icmp error-interval 1000000 bucket-size
20
```

Ping for IPv6 (Ping6)

The Ping6 test is a point-to-point test that accepts an IPv6 address or IPv6 host name to see if an IPv6 switch is communicating properly with another device on the same or another IP network. A ping test checks the path between the switch and another device by sending IP packets (ICMP Echo Requests).

To use a **ping6** command with an IPv6 host name or fully qualified domain names, refer to "DNS Resolver for IPv6" on page 6-8.

You can issue single or multiple ping tests with varying repetitions and timeout periods to wait for a ping reply.

Replies to each ping test are displayed on the console screen. To stop a ping test before it finishes, press **[Ctrl] [C]**.

For more information about using a ping test, refer to the "Troubleshooting" appendix in the current *Management and Configuration Guide* for your switch.

Syntax: ping6 < *ipv6-address* | *hostname* | *switch-number* >
[repetitions < 1 - 10000 >] [timeout < 1 - 60 >] [source <ipv6-address> | <vlan-id>] [data-size < 0 - 65507 >] [data-fill < 0 - 1024 >]
ping6 <*link-local-address*%vlan<*vid*> | *hostname* | *switch-number*>
[repetitions < 1 - 10000 >] [timeout < 1 - 60 >] [source <ipv6-address> | <vlan-id>] [data-size < 0 - 65507 >] [data-fill < 0 - 1024 >]

Pings the specified IPv6 host by sending ICMP version 6 (ICMPv6) echo request packets to the specified host.

<*ipv6-address*>: *IPv6 address of a destination host device.*

< *link-local-address* >%vlan<*vlan-id*>: *IPv6 link-local address, where* **%vlan**<*vlan-id*> *specifies the VLAN ID number.*

< *hostname* >: *Host name of an IPv6 host device configured on an IPv6 DNS server.*

< *switch-number* >: *Number of an IPv6-based switch that is a member of a switch stack (IPv6 subnet). Valid values: 1 - 16.*

[repetitions]: *Number of times that IPv6 ping packets are sent to the destination IPv6 host. Valid values: 1 - 10000. Default: 1.*

[timeout]: *Number of seconds within which a response is required from the destination host before the ping test times out. Valid values: 1 - 60. Default: 1 second.*

[source <ipv6-addr | hostname >]: *Source IP address or hostname. The source IP address must be owned by the router. If a VLAN is specified, the IP address associated with the specified VLAN is used.*

[data-size]: *Size of data (in bytes) to be sent in ping packets. Valid values: 0 - 65507. Default: 0.*

[data-fill]: *Text string used as data in ping packets. You can enter up to 1024 alphanumeric characters in the text.* ***Default:*** **0** *(no text is used).*

```
HP Switch# ping6 fe80::2:1%vlan10
fe80:0000:0000:0000:0000:0000:0002:0001 is alive, time = 975 ms

HP Switch# ping6 2001:db8::a:1c:e3:3 repetitions 3
2001:0db8:0000:0000:000a:001c:00e3:0003 is alive, iteration 1, time = 15 ms
2001:0db8:0000:0000:000a:001c:00e3:0003 is alive, iteration 2, time = 15 ms
2001:0db8:0000:0000:000a:001c:00e3:0003 is alive, iteration 3, time = 15 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms) min/avg/max = 15/15/15

HP Switch# ping6 2001:db8::214:c2ff:fe4c:e480 repetitions 3 timeout 2
2001:db8:0000:0000:0214:c2ff:fe4c:e480 is alive, iteration 1, time = 15 ms
2001:db8:0000:0000:0214:c2ff:fe4c:e480 is alive, iteration 2, time = 10 ms
2001:db8:0000:0000:0214:c2ff:fe4c:e480 is alive, iteration 3, time = 15 ms

HP Switch# ping6 2001:db8::10
Request timed out.
```

Figure 6-1. Examples of IPv6 Ping Tests

Traceroute for IPv6

The **traceroute6** command enables you to trace the route from a switch to a host device that is identified by an IPv6 address or IPv6 host name. In the command output, information on each (router) hop between the switch and the destination IPv6 address is displayed.

To use a **traceroute6** command with an IPv6 host name or fully qualified domain names, refer to "DNS Resolver for IPv6" on page 6-8.

Note that each time you perform a traceroute operation, the **traceroute** command uses the default settings unless you enter different values with each instance of the command.

Replies to each traceroute operation are displayed on the console screen. To stop a traceroute operation before it finishes, press **[Ctrl] [C]**.

For more information about how to configure and use a traceroute operation, refer to the "Troubleshooting" appendix in the *Management and Configuration Guide*.

Syntax: traceroute6 < *ipv6-address* | *hostname* >
[minttl < 1-255 > [maxttl < 1-255 > [timeout < 1 - 60 >] [probes < 1-5 >] [source <ipv6-addr | vlan-id>]

traceroute6 <*link-local-address*%vlan<*vid*> | *hostname* >
[minttl < 1-255 >] [maxttl < 1-255 >] [timeout < 1 - 60 >] [probes < 1-5 >] [source <ipv6-addr | vlan-id>]

Displays the IPv6 address of each hop in the route to the specified destination host device with the time (in microseconds) required for a packet reply to be received from each next-hop device.

<*ipv6-address*>: *IPv6 address of a destination host device.*

<*link-local-address*>%vlan<*vlan-id*>: *IPv6 link-local address, where* **%vlan<*vlan-id*>** *specifies the VLAN ID number.*

<*hostname*>: *Host name of an IPv6 host device configured on an IPv6 DNS server.*

minttl: *Minimum number of hops allowed for each probe packet sent along the route.* ***Default:*** *1;* ***Range:*** *1 - 255.*

- *If the* **minttl** *value is greater than the actual number of hops, the traceroute output displays only the hops equal to or greater than the configured* **minttl** *threshold value. The hops below the threshold value are not displayed.*
- *If the* **minttl** *value is the same as the actual number of hops, only the final hop is displayed in the command output.*
- *If the* **minttl** *value is less than the actual number of hops, all hops to the destination host are displayed.*

maxttl: *Maximum number of hops allowed for each probe packet sent along the route. Valid values: 1 - 255. Default: 30.*

- *If the* **maxttl** *value is less than the actual number of hops required to reach the host, the traceroute output displays only the IPv6 addresses of the hops detected by the configured* **maxttl** *value.*

timeout: *Number of seconds within which a response is required from the IPv6 device at each hop in the route to the destination host before the traceroute operation times out.* ***Default:*** *5 seconds;* ***Range:*** *1 - 60.*

probes: *Number of times a traceroute is performed to locate the IPv6 device at any hop in the route to the specified host before the operation times out.* ***Default:*** *3;* ***Range:*** *1 - 5.*

source <ipv6-addr | vlan-id>: *The source IP address or VLAN. The source IP address must be owned by the router. If a VLAN is specified, the IP address associated with the specified VLAN is used*

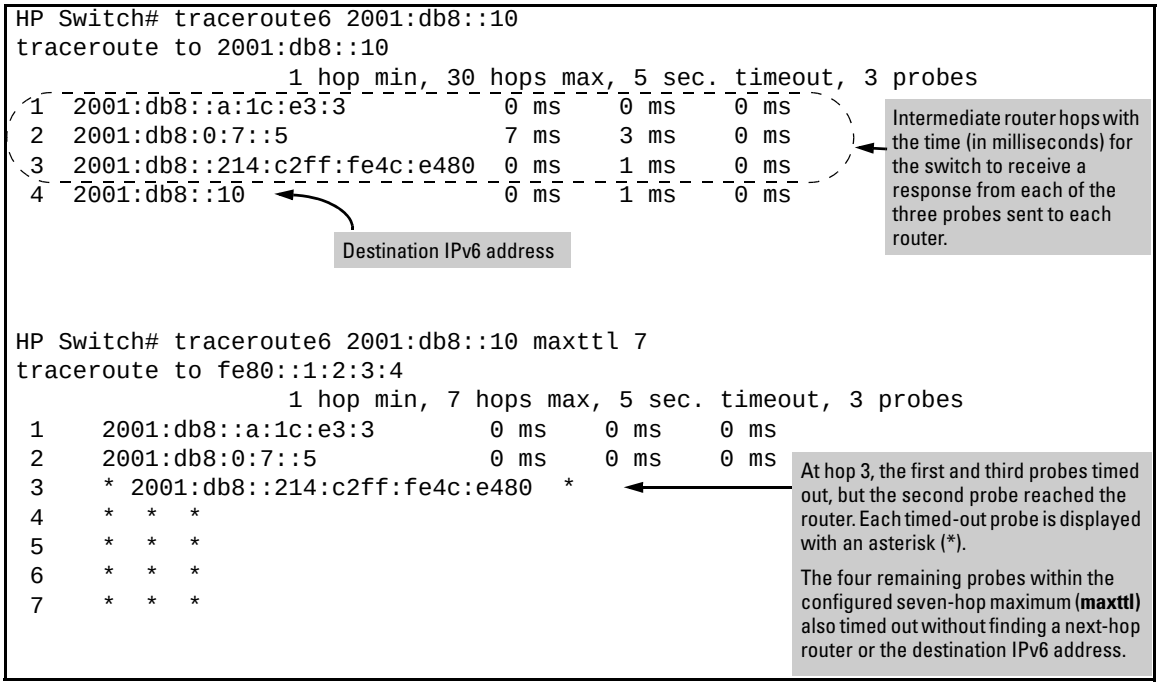


Figure 6-2. Examples of IPv6 Traceroute Probes

DNS Resolver for IPv6

The Domain Name System (DNS) resolver is designed for local network domains where it enables use of a host name or fully qualified domain name to support DNS-compatible commands from the switch. DNS operation supports these features:

- dual-stack operation: IPv6 and IPv4 DNS resolution
- DNS-compatible commands: **ping**, **ping6**, **traceroute**, and **traceroute6**
- multiple, prioritized DNS servers (IPv4 and IPv6)

DNS Configuration

Up to three DNS servers can be configured. The addresses must be prioritized, and can be for any combination of IPv4 and IPv6 DNS servers.

Note

This section describes the commands for configuring DNS operation for IPv6 DNS applications. For further information and examples on using the DNS feature, refer to "DNS Resolver" in appendix C, "Troubleshooting", in the current *Management and Configuration Guide* for your switch.

Syntax:. [no] ip dns server-address priority < 1 - 3 > < *ip-addr* >

Used at the global config level to configure the address and priority of a DNS server. Allows for configuring up to three servers providing DNS service. (The servers must all be accessible to the switch.) The command allows both IPv4 and IPv6 servers in any combination and any order of priority.

priority < 1 - 3 >: *Identifies the order in which the specified DNS server will be accessed by a DNS resolution attempt. A resolution attempt tries each configured DNS server address, in ascending order of priority, until the attempt is successful or all configured server options have been tried and failed. To change the priority of an existing server option, you must remove the option from the switch configuration and re-enter it with the new priority. If another server address is configured for the new priority, you must also remove that address from the configuration before re-assigning its priority to another address.*

— Continued on the next page. —

— Continued from the previous page. —

The **no** *form of the command removes the specified address from the server address list configured on the switch.*

< *ip-addr*>: *Specifies the address of an IPv6 or IPv4 DNS server.*

Syntax:. [no] ip dns domain-name < *domain-name-suffix* >

Used at the global config level to configure the domain suffix *that is automatically appended to the host name entered with a command supporting DNS operation. Configuring the domain suffix is optional if you plan to use fully qualified domain names in all cases instead of just entering host names.*

You can configure up to three addresses for DNS servers in the same or different domains. However, you can configure only one domain name suffix. This means that a fully qualified domain name must be used to resolve addresses for hosts that do not reside in the same domain as the one you configure with this command. That is, if the domain name suffix and the address of a DNS server for that same domain are both configured on the switch, then you need to enter only the host name of the desired target when executing a command that supports DNS operation. But if the DNS server used to resolve the host name for the desired target is in a different domain than the domain configured with this command, then you need to enter the fully qualified domain name for the target.

The **no** *form of the command removes the configured domain name suffix.*

For example, suppose you want to configure the following on the switch:

- the address **2001:db8::127:10** which identifies a DNS server in the domain named mygroup.procurve.net
- a priority of 1 for the above server
- the domain suffix **mygroup.procurve.net**

Assume that the above, configured DNS server supports an IPv6 device having a host name of “mars-1” (and an IPv6 address of fe80::215:60ff:fe7a:adc0) in the “mygroup.procurve.net” domain. In this case you can use the device's host name alone to ping the device because the mygroup.procurve.net domain has

been configured as the domain name on the switch and the address of a DNS server residing in that domain is also configured on the switch. The commands for these steps are as follows:

```
HP Switch(config)# ip dns server priority 1 2001:db8::127:10
HP Switch(config)# ip dns domain-name mygroup.procurve.net
HP Switch(config)# ping6 mars-1
fe80::215:60ff:fe7a:adc0 is alive, time = 1 ms
```

Figure 6-3. Example of Configuring for a Local DNS Server and Pinging a Registered Device

However, for the same "mars-1" device, if mygroup.procurve.net was not the configured domain name, you would have to use the fully qualified domain name for the device named mars-1:

```
HP Switch# ping6 mars-1.mygroup.procurve.net
```

For further information and examples on using the DNS feature, refer to "DNS Resolver" in appendix C, "Troubleshooting", in the current *Management and Configuration Guide* for your switch.

Viewing the Current Configuration

Use the **show ip dns** command to view the current DNS server configuration.

Use the **show run** command to view both the current DNS server addresses and the current DNS domain name in the active configuration.

Operating Notes

DNS addressing is not configurable from a DHCPv6 server.

Debug/Syslog for IPv6

The Debug/System logging (*Syslog*) for IPv6 feature provides the same logging functions as the IPv4 version, allowing you to record IPv4 and IPv6 Event Log and debug messages on a remote device to troubleshoot switch or network operation. For example, you can send messages about routing misconfigurations and other network protocol details to an external device, and later use them to debug network-level problems.

Configuring Debug and Event Log Messaging

To specify the types of debug and Event Log messages that you want to send to an external device:

- Use the **debug** < *debug-type* > command to send messaging reports for the following types of switch events:
  - ACL “deny” matches
  - DHCP snooping events
  - Dynamic ARP protection events
  - Events recorded in the switch’s Event Log
  - IPv4 and RIP routing events
  - IPv6 DHCPv6 client and Neighbor Discovery events
  - LLDP events
- Use the **logging** < **severity** *severity-level* | **system-module** *system-module*> command to select a subset of Event Log messages to send to an external device for debugging purposes according to:
  - Severity level
  - System module

Debug Command

Syntax: [no] debug < *debug-type* >

Configures the types of IPv4 and IPv6 messages that are sent to Syslog servers or other debug destinations, where <*debug-type*> *is any of the following event types:*

acl

When a match occurs on an ACL "deny" statement with a **log** *parameter, an ACL message is sent to configured debug destinations. (Default: Disabled - ACL messages for traffic that matches "deny" entries are not sent.)*

all

Configures all IPv4 and IPv6 debug message types to be sent to configured debug destinations. (Default: Disabled - No debug messages are sent.)

arp-protect

Configures messages for Dynamic ARP Protection events to be sent to configured debug destinations. (Default: Disabled - No debug messages are sent.)

event

Configures Event Log messages to be sent to configured debug destinations.

Event Log messages are enabled to be automatically sent to debug destinations in the following conditions:

- *If no Syslog server address is configured and you enter the* **logging** *command to configure a destination address.*
- *If at least one Syslog server address is configured in the startup configuration and the switch is rebooted or reset.*

Event log messages are the default type of debug message sent to configured debug destinations.

ip

Configures IPv4 RIP routing messages to be sent to configured debug destinations.

Syntax:. [no] debug < *debug-type* > *(Continued)*

ip [rip < database | event | trigger >

Configures specified IPv4 RIP message types to be sent to configured debug destinations:
database— *Database changes*
event— *RIP events*
trigger— *Trigger messages*

ipv6

Configures messages for IPv6 DHCPv6 client and neighbor discovery events to be sent to configured debug destinations.

ipv6 [dhcpv6-client <events | packets> | nd]

Configures one of the following IPv6 message types to be sent to configured debug destinations:
dhcpv6-clients events — *DHCPv6 client events*
dhcpv6-clients packets — *Statistics on DHCPv6 packets transmitted on a switch configured as a DHCPv6 client*
nd— *Events during IPv6 neighbor discovery*

lldp

Configures all LLDP message types to be sent to configured debug destinations.

Configuring Debug Destinations

A Debug/Syslog destination device can be a Syslog server (up to six maximum) and/or a console session:

- Use the **debug destination** < **logging** | **session** | **buffer** > command to enable (and disable) Syslog messaging on a Syslog server or to a CLI session for the debug message types configured with the **debug** and **logging** commands (see "Configuring Debug and Event Log Messaging" on page 6-11):
  - **debug destination logging** enables the configured debug message types to be sent to Syslog servers configured with the **logging** command.
  - **debug destination session** enables the configured debug message types to be sent to the CLI session that executed this command. The session can be on any one terminal emulation device with serial, Telnet, or SSH access to the CLI at the Manager level prompt.
  - **debug destination buffer** enables the configured debug message types to be sent to a buffer in switch memory.

Logging Command

Syntax: [no] logging < *syslog-ipv4-addr* | *syslog-ipv6-addr* >

Enables or disables Syslog messaging to the specified IPv4 address. You can configure up to six addresses. If you configure an address when none are already configured, this command enables destination logging (Syslog) and the Event debug type. Therefore, at a minimum, the switch begins sending Event Log messages to configured Syslog servers. If other debug message types are configured, they are also sent to the Syslog server.

no logging *removes all currently configured Syslog logging destinations from the running configuration.*

no logging < *syslog-ipv4-address* > *removes only the specified Syslog logging destination from the running configuration.*

Note: *The* **no logging** *command does not delete the Syslog server addresses stored in the startup configuration. To delete Syslog addresses in the startup configuration, you must enter the* **no logging** *command followed by the* **write memory** *command. To verify the deletion of a Syslog server address, display the startup configuration by entering the* **show config** *command.*

To block the messages sent to configured Syslog servers from the currently configured debug message type, enter the **no debug** < *debug-type* > *command.*

To disable Syslog logging on the switch without deleting configured server addresses, enter the **no debug destination logging** *command.*

For complete information on how to configure a Syslog server and Debug/Syslog message reports, refer to the “Troubleshooting” appendix in the *Management and Configuration Guide.*

Terminology

DAD	Duplicate Address Detection. Refer to “Duplicate Address Detection (DAD)” on page 1-16.
Device Identifier	The low-order bits in an IPv6 address that identify a specific device. For example, in the link-local address 2001:db8:a10:101:212:79ff:fe88:a100/64, the bits forming 212:79ff:fe88:a100 comprise the device identifier.
DoS	Denial-of-Service.
EUI-64	Extended Unique Identifier.
Manual Address Configuration	Configures an IPv6 address by using the CLI to manually enter a static address. Referred to as “Static Address Configuration” in this guide. See **Static Address Configuration**, below.
MLD	Multicast Listener Discovery. Refer to the chapter titled “Multicast Listener Discovery (MLD) Snooping”.
MTU	Maximum Transmission Unit. The largest frame size allowed on a given path or device.
RA	Router Advertisement. Refer to “Router Advertisements” on page 1-26.
SLAAC	Stateless Address Autoconfiguration.
Static Address	A permanently configured IPv6 address, as opposed to an autoconfigured address.
Static Address Configuration	Configures an IPv6 address by using the CLI to manually enter the address instead of using an automatically generated or DHCPv6-assigned address. Same as “Manual Address Configuration”. See also **Manual Address Configuration**, above.

Index

Symbols

A

F

G

I

L

M

N

O

P

R

S

T

V

W

Technology for better business outcomes

To learn more, visit www.hp.com/networking

May 2013

Manual Part Number
5998-4256