Aruba 2540 IPv6 Configuration Guide for AOS-S 16.08

a Hewlett Packard Enterprise company

Part Number: 5200-5475
Published: December 2018
Edition: 1

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise 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. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Pentium®, Xeon®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the U.S. and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

This guide provides information on the IPv6 protocol information that are supported on the switch.

Applicable products

This guide applies to these products:

Aruba 2540 Switch Series (JL354A, JL355A, JL356A, JL357A)

Switch prompts used in this guide

Examples in this guide are representative and may not match your particular switch/environment. Examples use simplified prompts as follows:

Prompt	Explanation
`switch#`	`#` indicates manager context (authority).
`switch>`	`>` indicates operator context (authority).
`switch(config)#`	`(config)` indicates the config context.
`switch(vlan-x)#`	`(vlan-x)` indicates the vlan context of config, where *x* represents the VLAN ID. For example: `switch(vlan-128)#`.
`switch(eth-x)#`	`(eth-x)` indicates the interface context of config, where `x` represents the interface. For example: `switch(eth-48)#`.
`switch-Stack#`	`Stack` indicates that stacking is enabled.
`switch-Stack(config)#`	`Stack(config)` indicates the config context while stacking is enabled.
`switch-Stack(stacking)#`	`Stack(stacking)` indicates the stacking context of config while stacking is enabled.
`switch-Stack(vlan-x)#`	`Stack(vlan-x)` indicates the vlan context of config while stacking is enabled, where *x* represents the VLAN ID. For example: `switch-Stack(vlan-128)#`.
`switch-Stack(eth-x/y)#`	`Stack(eth-x/y)` indicates the interface context of config, in the form `(eth-<member-in-stack>/<interface>)`. For example: `switch(eth-1/48)#`

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 is 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.)

General configuration steps

This section provides an overview of the general configuration steps for enabling IPv6 on a given VLAN 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.

Procedure

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 generate automatically, 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, do either of the following:

   a. Enable IPv6 on the VLAN.

      Automatically configures a link-local address with an EUI-64 interface identifier.

   b. Statically configure a unicast IPv6 address on the VLAN.

      If you configure anything other than a link-local address, it enables IPv6 on the VLAN and the link-local address is automatically configured with an EUI-64 interface identifier

3. If an IPv6 router is connected on the VLAN, then enable IPv6 address autoconfiguration to configure automatically a global unicast address with prefixes included in advertisements received from the router. The

interface identifier used in addresses configured by this method will be the same as the interface identifier in the current link-local address.

4. If needed, statically configure IPv6 unicast addressing on the VLAN interface as needed. Also includes statically replacing the automatically generated link-local address.

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`

  (See **Enabling IPv6 with an automatically configured link-local address** on page 10.)

- `ipv6 address autoconfig`

  (See **Enabling autoconfiguration of a global unicast address and a default router identity on a VLAN** on page 11.)

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

  (See **Enabling DHCPv6** on page 12.)

- `ipv6 address fe80:0:0:0: <interface-identifier> link-local`

  (See **Statically configuring a link-local unicast address** on page 14.)

- `ipv6 address <prefix:interface-identifier>`

  (See **Statically configuring a global unicast address** on page 14.)

NOTE: Addresses created by any of these methods remain tentative until verified as unique by Duplicate Address Detection (DAD). See **Duplicate Address Detection (DAD)** on page 18.

Enabling IPv6 with an automatically configured link-local address

Syntax:

`ipv6 enable`

`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 and automatically configures the VLAN link-local unicast address with a 64-bit EUI-64 interface identifier

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 RAs to designate the default IPv6 router. See **Default IPv6 router** on page 24.

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

If no other IPv6-enabling command is configured on the VLAN, the `no` form of the command disables IPv6 on the VLAN. See **Disabling IPv6 on a VLAN** on page 15.

Viewing currently configured IPv6 unicast addresses

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 commands:

- `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, see **Viewing the current IPv6 addressing configuration** on page 19.

Enabling autoconfiguration of a global unicast address and a default router identity on a VLAN

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

Syntax:

```
ipv6 address autoconfig
no ipv6 address autoconfig
```

Implements unicast address autoconfiguration as follows:

- If IPv6 is not already enabled on the VLAN, enables IPv6 and generates a link-local EUI-64 address.
- Generates router solicitations (RS) on the VLAN.
- If an RA is received on the VLAN, the switch uses the route prefix in the RA to configure a global unicast address. Interface identifier for this address is the same as the interface identifier used in the current link-local address at the time the RA is received. It can be either a statically configured or the (automatic) EUI-64 interface 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 is present, but no global unicast addresses are autoconfigured.

  NOTE: If a link-local address is already configured on the VLAN, a later autoconfigured global unicast address will use the same interface 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, only the first to acquire a global unicast address is used.

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 enabled only by the `autoconfig` command, deleting this command disables IPv6 on the VLAN.

Viewing current IPv6 autoconfiguration settings

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

Syntax:

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

`show ipv6`

Lists IPv6 addresses for all VLANs configured on the switch.

Syntax:

`show ipv6 vlan <vid>`

Lists IPv6 addresses configured on the VLAN.

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 necessary to provide a global unicast address to a switch interface, the server can still be configured to provide the NTP server assignment. It is sometimes referred to as “stateless DHCPv6”.)

Syntax:

`ipv6 address dhcp full [rapid-commit]`

`no ipv6 address dhcp full [rapid-commit]`

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 enables IPv6 and causes the switch to autoconfigure a link-local unicast address with an EUI-64 interface identifier.

`[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).

NOTE: A DHCPv6 server does not assign link-local addresses, and enabling DHCPv6 on a VLAN does not affect a pre-existing link-local address.

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 the switch is an IPv6 host, `ipv6 address dhcp full` must be configured on the DHCPv6 client to obtain relevant information from the DHCPv6 server. M-bit and O-bit settings in RAs from a router are not used by the switch in host mode. If the switch is operating as an IPv6 router, it includes M-bit and O-bit values in the RAs it transmits. See **IPv6 Router Advertisements (RAs)** on page 118 for routing switch operation.

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.)

An IPv6 address assigned to the VLAN by a DHCPv6 server is set to the preferred and valid lifetimes specified in an RA received on the VLAN for the prefix used in the assigned address. It is configured as a preferred address after verification of uniqueness by DAD.

Default: Disabled

If no other IPv6-enabling command is configured on the VLAN, the `no` form of the command removes the DHCPv6 option from the configuration and disables IPv6 on the VLAN See **Disabling IPv6 on a VLAN** on page 15.

Viewing configured DHCPv6 addresses

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

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

Syntax:

`show ipv6`

Lists IPv6 addresses for all VLANs configured on the switch.

Syntax:

`show running-config vlan <VLAN ID>`

Lists IPv6 addresses configured on the VLAN.

For more information, see **Viewing the current IPv6 addressing configuration** on page 19.

General operating notes for DHCPv6

- 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.
- DHCPv6 authentication and stateless DHCPv6 are not supported in this software.
- With IPv6 enabled, the switch determines the default IPv6 router for the VLAN from the RAs it receives. See **Default IPv6 router** on page 24.
- 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 is not 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, an autoconfigured address assignment replaces it after the next RA is received on the VLAN. DHCPv6 and autoconfigured addresses coexist on the same VLAN if they belong to different subnets.

For related information, see:

- 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 interface identifiers)
- global unicast (and unique local unicast)

Statically configuring a link-local unicast address

Syntax:

`ipv6 address fe80::<interface-id> link-local`

`no ipv6 address fe80::<interface-id> 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, this command overwrites the current, link-local address with the specified static address. (One link-local address is allowed per VLAN interface.)

<interface-id>

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 interface identifier as the link-local address.

NOTE: 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, which 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 enabled only by a statically configured link-local address, deleting the link-local address disables IPv6 on the VLAN.
- If other IPv6-enabling commands have been configured on the VLAN, 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.

See also **Disabling IPv6 on a VLAN** on page 15.

Statically configuring a global unicast address

Syntax:

`ipv6 address [<network-prefix> <interface-id> | <prefix-length> ]`

`no ipv6 address [<network-prefix> <interface-id> | <prefix-length> ]`

`ipv6 address [<network-prefix> ::/ <prefix-length> eui-64 ]`

`no ipv6 address [<network-prefix> ::/ <prefix-length> eui-64 ]`

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

If IPv6 is not already enabled on a VLAN, either of these command options enable IPv6 on the VLAN, configure a link-local address using the EUI-64 format, or statically configure a global unicast 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.

If no other IPv6-enabling command is configured on the VLAN, the `no` form of the command erases the specified address and disables IPv6 on the VLAN.

<network-prefix> : Includes the global routing prefix and the subnet ID for the address.

<interface-id> : Enters a user-defined interface 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 (EUI) format to create an interface identifier based on the VLAN MAC address.

Viewing the currently configured static IPv6 addresses per-VLAN

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

Syntax:

```
show ipv6
```

Lists IPv6 addresses for all VLANs configured on the switch.

Syntax:

```
show ipv6 vlan <vid>
```

Lists IPv6 addresses configured on `VLAN <vid>` .

For more information, see **Viewing the current IPv6 addressing configuration** on page 19.

Operating notes for DHCPv6

- With IPv6 enabled, the switch determines the default IPv6 router for the VLAN from the RAs it receives.
- If DHCPv6 is configured on a VLAN, then configuring a static global unicast address on the VLAN removes DHCPv6 from the configuration of the VLAN and deletes the DHCPv6-assigned global unicast address.
- For a statically configured global unicast address to be routable, a gateway router must be transmitting RAs on the VLAN.
- If an autoconfigured global unicast address 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.

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, the more recent, duplicate address is designated as `duplicate`. For more on this topic, see:

- **Duplicate Address Detection (DAD)** on page 18
- **Viewing the current IPv6 addressing configuration** on page 19

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, the following commands must be removed from the VLAN configuration:

```
ipv6 enable
ipv6 address dhcp full [rapid-commit]
ipv6 address autoconfig
ipv6 address fe80::<interface-id> link-local
ipv6 address <prefix>:<interface-id>
```

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

Neighbor Discovery (ND)

Neighbor Discovery (ND) is the IPv6 equivalent of the IPv4 ARP for layer 2 address resolution. ND uses IPv6 ICMP messages to provide for discovery of IPv6 devices such as other switches, routers, management stations, and servers on the same interface. ND runs automatically in the default configuration and provides services along with those provided in IPv4 by ARP. For example:

- 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)
- RA processing
- neighbor reachability
- autoconfiguration of unicast addresses
- resolution of destination addresses
- changes to link-layer addresses

An instance of ND is triggered on a device when a new (tentative) or changed IPv6 address is detected. (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. ND 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 an RA.

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, see RFC 2460.

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 time after communication ceases and then dropped.

To view or clear the content of the neighbor cache, see **Viewing the Neighbor Cache** on page 28.

For related information, see RFC 2461: "Neighbor Discovery for IP Version 6 (IPv6)."

DHCPv6 client

The DHCPv6 client allows a host to request global unicast IPv6 address assignments from a DHCPv6 server. If there are multiple DHCPv6 servers, the client can select a server based on the preference value sent in DHCPv6 messages.

The DHCPv6 client can request that the server send only the configuration information. In this case, a router on the same interface (VLAN) as the host provides the global IPv6 address to the host through router advertisements.

NOTE:

If the switch is rebooted with a default configuration, only the default DHCPv4 client is enabled on the default VLAN. The DHCPv6 client has to be explicitly enabled on a VLAN using the command `ipv6 address dhcp` or `ipv6 address autoconfig`.

ipv6 dhcp-client

Syntax

```
ipv6 dhcp-client authentication mode <auth_mode> | key-chain <key_chain_name>

no ipv6 dhcp-client authentication mode <auth_mode> | key-chain <key_chain_name>
```

Description

Configures DHCPv6 clients authentication mode and key chain.

The `no` form of this command disables the DHCPv6 client authentication.

Command context

```
vlan-x
```

Parameters

<auth_mode>

Specify the type of authentication to be used.

<key_chain_name>

Specify the key chain to be used for authentication.

Examples

```
switch(vlan-x)# ipv6 dhcp-client authentication
key-chain             Specify the key chain to be used for authentication.
mode                  Specify the type of authentication to be used.
switch(vlan-x)# ipv6 dhcp-client authentication key-chain
CHAIN-NAME-STR        The name of the key chain.
switch(vlan-x)# ipv6 dhcp-client authentication key-chain dhcp10
switch(vlan-x)# ipv6 dhcp-client authentication mode
md5                   Set the authentication mode to MD5.
switch(vlan-x)# ipv6 dhcp-client authentication mode md5
```

```
switch(config)# show ipv6 dhcp-client authentication

DHCPv6 Authentication Information

Vlan Name            : DEFAULT_VLAN
Authentication       : Enabled
Authentication Mode  : md5
Key-Chain            : dhcp10
Key-Id               : 1
```

```
switch(config)# show ipv6 dhcp-client authentication vlan 1

DHCPv6 Authentication Information

Vlan Name            : DEFAULT_VLAN
Authentication       : Enabled
Authentication Mode  : md5
Key-Chain            : dhcp10
Key-Id               : 1
```

Duplicate Address Detection (DAD)

DAD 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 or per-interface command level. DAD can be useful in helping to troubleshoot erroneous replies to DAD requests, or where the neighbor cache contains many invalid entries caused by an unauthorized station sending false replies to the ND queries of the switch. 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, 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 labeled 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 a similar 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 previously.

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 <0-255>`

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.

`<0-255>` : The number of consecutive neighbor solicitation messages sent for DAD inquiries on an interface. Setting this value to 0 disables DAD on the interface, which 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: 0–255 (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 ND 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 an RA to help ensure that all hosts on a VLAN are using the same retransmit interval for ND. See **Setting or changing the hop-limit for host-generated packets** on page 123.

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

Default: 1000 ms; Range: 1000–4294967295 ms

Syntax:

`ipv6 nd reachable-time <milliseconds>`

Used on VLAN interfaces to configure the length of time in milliseconds a neighbor is 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 an RA.

This value can be configured in an RA to help ensure that all hosts on a VLAN are using the same reachable time in their neighbor cache.

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

Default: 30,000 ms; Range: 1000–3600000 ms

Operating notes for ND

- 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 DAD.
- 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 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.

Viewing the current IPv6 addressing configuration

Use these commands to view the 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` : Global setting and is not configured per-VLAN.

`Default Gateway` : Lists the IPv4 default gateway, if any, configured on the switch.It 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 ND.

`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 (such as after a reboot). Default: 3; Range: 0–255 ms. A setting of “0” disables duplicate address detection. See **Duplicate Address Detection (DAD)** on page 18.

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

`IPv6 Status` : For the indicated VLAN, shows whether IPv6 is disabled (the default) or enabled. See **Configuring IPv6 addressing** on page 10.

`Address Origin` :

`Autoconfig` : The address was configured using stateless address autoconfiguration (SLAAC). In this case, the interface identifier for global unicast addresses copied from the current link-local unicast address.

`DHCP` : The address is assigned by a DHCPv6 server. 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 it 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. See **Address lifetimes** on page 26.

`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 exists on another device belonging to the same VLAN interface. A duplicate address is not used.

The display below shows the output for a switch having IPv6 enabled on one VLAN.

```
Switch# show ipv6

Internet (IPv6) Service

  IPv6 Routing    : Enabled
  Default Gateway : fe80::213:c4ff:fedd:14b0
  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
  ---------- + ------------------------------------ ---------
  dhcp       | 2001:db8:a03:e102::1:101/64          preferred
  manual     | fe80::1:101/64                       preferred
```

```
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 | 2930:0:a03:e102::127/64               preferred
  dhcp       | 2930:0:a03:e102:212:79ff:fe88:a100/64 preferred
  manual     | fe80::127/64                          preferred
```

Syntax:

`show ipv6 nd`

Displays the current IPv6 ND settings on the configured VLAN interfaces.

For example, the display below shows the output for a switch having IPv6 enabled on VLANs 1 and 22.

```
Switch# show ipv6 nd

IPV6 Neighbor Discovery Configuration

  VLAN ID   DAD       RCH Time   NS Interval
            Attempts  (msecs)    (msecs)
  --------- --------- ---------- -----------
  1         3         30000      1000
  22        3         30000      1000
```

Syntax:

`show ipv6 vlan <vid>`

Displays IPv6 addresses 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`

  (See **Router access and default router selection** on page 23.)

- `default gateway`

  Lists the IPv4 default gateway, if any, configured on the switch.It is a globally configured router gateway address and is not configured per-VLAN

- `nd dad`

Shows whether ND is enabled. The default setting is Enabled. Using `ipv6 nd dad- attempts 0` disables ND.

- `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 (such as after a reboot).Default: 3; Range: 0–255 ms. A setting of “0” disables duplicate address detection. See **Duplicate Address Detection (DAD)** on page 18.

- `VLAN Name`

  Lists the name of a VLAN statically configured on the switch.

- `IPv6 Status`

  For the indicated VLAN, shows whether IPv6 is disabled (the default) or enabled. See **Configuring IPv6 addressing** on page 10.

- `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 are specified in the RA 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.

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

Internet (IPv6) Service

  IPv6 Routing    : Disabled
  Default Gateway : fe80::213:c4ff:fedd:14b0%vlan10
  ND DAD          : Enabled
  DAD Attempts    : 3

  Vlan Name       : VLAN10
  IPv6 Status     : Enabled

  IPv6 Address/Prefixlength         Expiry
  --------------------------------- ------------------------
  2001:db8:a03:e102::1:101/64       Fri May 19 11:51:15 2014
  fe80::1:101/64                    permanent
```

Syntax:

`show run`

Along with 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 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::<interface-id> link-local
   ipv6 address <prefix>:<interface-id>/<prefix-length>
   ipv6 address autoconfig
   ipv6 address dhcp full [rapid-commit]
   ipv6 <global-unicast-address>/<prefix>
```

```
Switch(config)# show run

Running configuration:
. . .
vlan 10
   name "VLAN10"
   untagged 1-12
   ipv6 address fe80::1:101 link-local
   ipv6 address dhcp full rapid-commit
. . .
```

Statically configured IPv6 addresses appear in the `show run` output.

Commands for automatic IPv6 address configuration appear in the `show run` output, but the addresses resulting from these commands do not appear in the output.

Router access and default router selection

Traffic can be routed between destinations on different VLANs configured on the switch or to a destination on an off-switch VLAN. This 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 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 ND 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. (It 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 an 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 default router list and the switch stops sending router solicitations. If an RA is not received, 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 an RA after sending the router solicitations, no further router solicitations are sent on that VLAN unless a new IPv6 setting is configured. IPv6 on the VLAN is disabled and then re-enabled, or the up-links in VLAN are disconnected and reconnected.

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.

- If the switch receives 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. This is based on the relative reachability of the router, 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 one. For further information on routing IPv6 traffic, see the documentation provided for the IPv6 router.

For related information, see RFC 2461: "Neighbor Discovery for IP Version 6."

View IPv6 gateway, route, and router neighbors

Use these commands to view the current routing table content of the switch and connectivity to routers per VLAN. This includes information received in RAs from IPv6 routers on VLANs enabled with IPv6 on the switch. See **Viewing gateway and IPv6 route information**.

Viewing gateway and IPv6 route information

Syntax:

`show ipv6 route [`*`ipv6-addr`*`] [connected]`

Displays the routes in the IPv6 routing switch 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 or connected).

`Distance` : The administrative distance of the route used to determine the best path to the destination.

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

The following example displays output for `show ipv6 route` command:

```
Switch(config)# show ipv6 route

                          IPv6 Route Entries

Dest : ::/0                                                Type : static
Gateway : fe80::213:c4ff:fedd:14b0%vlan10    Dist. : 40    Metric : 0

Dest : ::1/128                                            Type : connected
Gateway : lo0                                Dist. : 0     Metric : 1

Dest : 2001:db8:a03:e102::/64                             Type : connected
Gateway : VLAN10                             Dist. : 0     Metric : 1

Dest : fe80::%vlan10                                     Type : connected
Gateway : VLAN10                             Dist. : 0     Metric : 1

Dest : fe80::1%lo0                                       Type : connected
Gateway : lo0                                Dist. : 0     Metric : 1
```

Viewing IPv6 router information

Syntax:

`show ipv6 routers [vlan <vid>]`

Lists the 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 RAs have been received on the switch.

NOTE: This command reports on IPv6 routers the switch has learned of through operation as a host or client of a router. If the switch itself is configured as an IPv6 router (routing switch), the output of this command is empty.

`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 router exists.

`MTU` : 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. See **Valid lifetime** on page 26.

`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. See **Preferred lifetime** on page 26.

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

The display below indicates that the switch is receiving RAs from a single router that exists on VLAN 10.

```
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
```

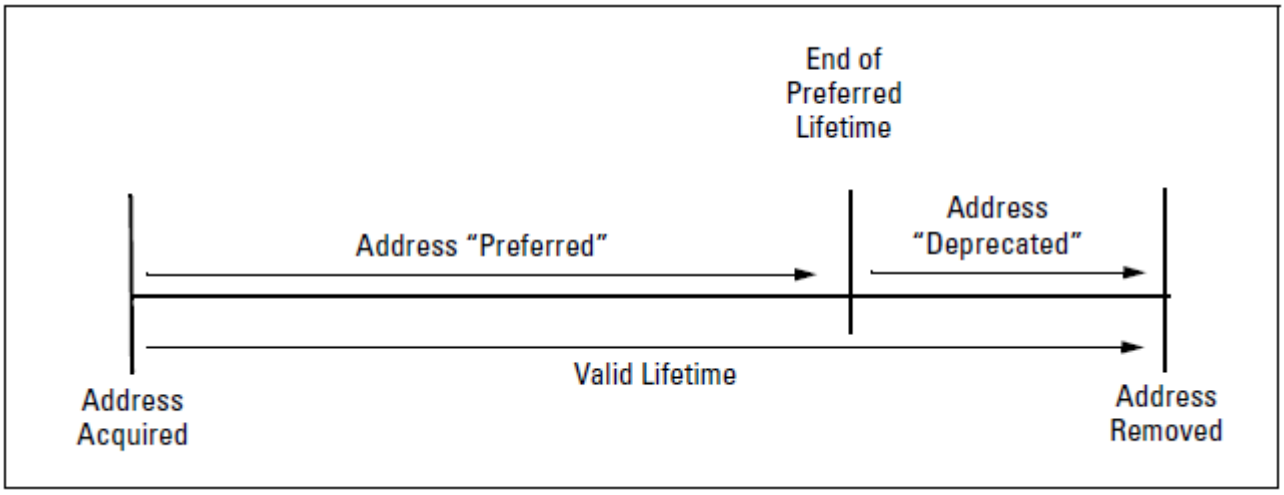
Address lifetimes

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 specifies the duration of their use and availability.

Preferred and valid address lifetimes

Autoconfigured IPv6 global unicast addresses acquire their valid and preferred lifetime assignments from RAs. A valid lifetime is the time period during which an address is allowed to remain available and usable on an interface. A preferred lifetime is the length of time an address is intended for full use on an interface, and must be less than or equal to the address's valid lifetime.

Figure 1: *Preferred and valid address lifetimes*



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 both appear in the `show ipv6 vlan <vid>` output as long as the deprecated address is within its valid lifetime.

Valid lifetime

The valid lifetime, which is the total time the address is available, 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 time frame, the address is no longer to 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. The following table lists the IPv6 unicast address 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

NOTE: Preferred and valid lifetimes on a VLAN interface are determined by the RAs received on the interface. These values are not affected by the lease time assigned to an address by a DHCPv6 server. That is, lease expiration on a DHCPv6-assigned address terminates use of the address, regardless of the status of the RA-assigned lifetime, and router-assigned lifetime expiration of a leased address terminates the switch's use of the address. (The router-assigned lifetime can be extended by receipt of a new RA.) Statically configured IPv6 addresses are regarded as permanent addresses, and do not expire.

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

Related Information:

- RFC 2462: "IPv6 Stateless Address Autoconfiguration"
- RFC 4291: "IP Version 6 Addressing Architecture"

IPv6 management features include viewing and clearing the IPv6 neighbor cache, telnet operations, configuring and enabling SNTP and Timep, TFTP file transfers over IPv6, SNMP management, and IP preserve for IPv6. For additional information on these features, see the current *ArubaOS-Switch Management and Configuration Guide* for your switch.

Viewing and clearing the IPv6 Neighbor 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.

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, see **Neighbor Discovery (ND)** on page 16.

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 the 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, 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:

INCMP: (Incomplete): Neighbor address resolution is in progress, but has not yet been determined.

REACH: (Reachable): The neighbor is known to have been reachable recently.

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 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.

The output for `show ipv6 neighbors` command is:

```
Switch(config)# show ipv6 neighbors
 IPv6 ND Cache Entries
 IPv6 Address                MAC Address   State Type    Port
 --------------------------- ------------- ----- ------- ----
 2001:db8:260:212::101       0013c4-dd14b0 STALE dynamic 1
 2001:db8:260:214::1:15      001279-88a100 REACH local
 fe80::1:1                   001279-88a100 REACH local
 fe80::10:27                 001560-7aadc0 REACH dynamic 3
 fe80::213:c4ff:fedd:14b0    0013c4-dd14b0 REACH dynamic 1
```

The output for neighbor cache content for a specific VLAN is:

```
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    1
 2001:db8:260:214::1:15    001279-88a100 REACH 11h:15m:23s   17
 fe80:1a3::1:1             001279-88a100 REACH 9h:35m:11s    12
 fe80:::10:27              001560-7aadc0 REACH 22h:26m:12s   3
 fe80::213:c4ff:fedd:14b0 0013c4-dd14b0 REACH 23 0h:32m:36s 1
```

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 many invalid entries and communication with some valid hosts may fail The `show ipv6 neighbors` command output also 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 nonlocal IPv6 neighbor addresses and corresponding MAC addresses from the neighbor cache, except neighbor entries specified as next-hops for active routes. The Layer-2 address information for any next-hop route is cleared until the route is refreshed in the neighbor cache. An example for clearing the IPv6 neighbors cache is:

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

Switch# show ipv6 neighbors

 IPv6 ND Cache Entries

 IPv6 Address                MAC Address   State Type    Port
 --------------------------- ------------- ----- ------- ----
 fe80::213:c4ff:fedd:14b0    000000-000000 INCMP dynamic
```

For an active-route next-hop, the MAC address and source port data is removed, and the State is set to “Incomplete” (INCMP) until the route is refreshed in the neighbor cache.

IPv6 Telnet operations

The IPv6 Telnet operation includes using outbound telnet to another device, viewing the current telnet activity on a switch, enabling or disabling inbound or outbound telnet access, and viewing the current inbound or outbound telnet operation. For IPv4 Telnet operation, see the *ArubaOS-Switch Management and Configuration Guide* for your switch.

Using outbound Telnet to another device

Syntax:

```
telnet <link-local-addr> %vlan <vid>

telnet <global-unicast-addr>
```

Outbound Telnet 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 RAs from an IPv6 gateway router.*<global–unicast–addr>*: Specifies the global IPv6 address of the destination device.

Telnet to another device

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), use the following command:

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

If the switch is receiving RAs 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.

To Telnet to a device having an IPv6 global unicast address of 2001:db8::215:60ff:fe79:980, enter the following command:

```
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 or outbound sessions.

Privilege: Manager or Operator.

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

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

Output for `show telnet` with three sessions active

```
Switch# show telnet

 Telnet Activity

 ------------------------------------------------------
 Session  : 1
 Privilege: Manager
 From     : Console
 To       : 10.0.10.140
 ------------------------------------------------------
 Session  : 2
 Privilege: Manager
 From     : 2540:0:260:212::2:219
 To       :
 ------------------------------------------------------
 Session  : ** 3
 Privilege: Manager
 From     : fe80::2:101
 To       :
```

The ** in “Session:” indicates the session through which `show telnet` was run.

This output shows that the switch is running one outbound IPv4 session and is being accessed by two inbound or outbound sessions.

Enabling or disabling inbound or outbound Telnet access

Syntax:

`telnet-server`

`no telnet-server`

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

For example, to disable IPv4 and IPv6 Telnet access to the switch, you would use this command:

`Switch(config)# no telnet-server`

Viewing the current inbound or outbound Telnet configuration

Syntax:

`show console`

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

Output for `show console` showing default console configuration

```
Switch(config)# show console

 Console/Serial Link

Active Consolec: Serial
USB Console Input Enabled [Yes] : Yes
inbound or outbound 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
Global Session Idle Timeout(sec) [0] : 0
Serial/USB Console Idle Timeout (sec) [not set/900]: not set
Current Session Idle Timeout (sec) : 0
Maximum Concurrent Sessions Allowed [7] : 7
Maximum Concurrent Sessions Allowed Per User [7] : 7
Switch(config)#
```

Inbound or outbound Telnet Setting for IPv4 and IPv6 Telnet is `Yes`.

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, see the *ArubaOS-Switch Management and Configuration Guide* for your switch.

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:

`sntp server priority <1-3> <link-local-addr> %vlan <vid> [1-7]`

`no sntp server priority <1-3> <link-local-addr> %vlan <vid> [1-7]`

Syntax:

`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 until 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.

Configuring link-local and global unicast SNTP server addresses

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.

```
Switch(config)# sntp server priority 1 fe80::215:60ff:fe7a:adc0%vlan10 7
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 proceeding `sntp server` command example would appear as follows:

Output for `show sntp` with both an IPv6 and an IPv4 server address configured

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

```
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
```

NOTE: The `show management` command can also be used to display SNTP server information.

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, see the *ArubaOS-Switch 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.
2. Enter a TFTP `copy` command with the IPv6 address of a TFTP server in the command syntax.
3. Optional: To enable auto-TFTP operation, enter the `auto-tftp` command.

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 [listen <data|both>]>
```

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

NOTE: 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.

The `no tftp <client|server>` command does not affect auto-TFTP operation. For more information, see **Using auto-TFTP for IPv6** on page 36.

Copying files over IPv6 using TFTP

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>` : One of the following values:

`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, see the *ArubaOS-Switch Access Security Guide* for your switch.

`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 <source> tftp <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 and then reboots the switch. To implement the process, the switch must first reboot using one of the following methods:

Procedure

1. Enter the **`boot system flash primary`** command in the CLI.
2. 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.

NOTE: To enable auto-TFTP to copy a software image to primary flash memory, the version number of the downloaded software file (for example, xx_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. See **Enabling TFTP for IPv6** on page 34 .

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

Supported SNMP features

The same SNMP for IPv4 features is 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
- Flow sampling using sFlow
- Standard MIBs, such as the Bridge MIB (RFC 1493) and the Ethernet MAU MIB (RFC 1515)

Supported SNMP configuration commands

For more information on each SNMP configuration procedure, see the current *ArubaOS-Switch 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 <params_name>
<ipv4-addr|ipv6-addr>
[ addr-mask <ipv4-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.

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> ]`

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.

Output of `show snmp-server` command with IPv6 address

```
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
  2540:0000:0260:0211
  :0217:a4ff:feff:1f70  marker    Critical trap 3     15

 Excluded MIBs

 Snmp Response Pdu Source-IP Information

 Selection Policy : rfc1517
```

Note: An IPv6 address is displayed on two lines.

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.

Output of `show snmpv3 targetaddress` command with IPv6 address

```
Switch(config)# show snmpv3 targetaddress

 snmpTargetAddrTable [rfc2573]

  Target Name      IP Address              Parameter
```

```
  ---------------- ---------------------- ----------
  1                15.29.17.218           1
  2                15.29.17.219           2
  PP.217           15.29.17.217           marker_p
  PP.218           2540:0:260:211
                      :217:a4ff:feff:1f70  marker_p
```

Note: An IPv6 Address is displayed on two lines

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 (the default VLAN) in each switch, and the Gateway IPv6 address assigned to the switch.

Configuring IP preserve

Enter the `ip preserve` statement at the end of the configuration file to be downloaded from a TFTP server. (You do not invoke IP preserve by entering a command from the CLI.)

How to enter IP preserve in a configuration file

```
; J9627A Configuration Editor; Created on release #xx.16.xx
hostname "Switch"
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.

IP preserve configuration file download to an IPv6-based switch

Enter the TFTP `copy` command 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.

Verifying how IP preserve was implemented in a switch

After the switch reboots, enter the `show run` command. The example below shows all the configurations settings that 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.

If a switch received its IPv6 address from a DHCP server, the "ip address" field under "vlan 1" would display `dhcp-bootp`.

Configuration file with dedicated IP addressing

```
Switch(config)# show run

Running configuration:

; J8715A Configuration Editor; Created on release #xx.14.01

hostname "Switch"
module 1 type J8702A
module 2 type J8705A
trunk 11-12 Trk1 Trunk
ipv6 default-gateway 2001:db8:0:7::5
snmp-server community "public" Unrestricted
vlan 1
   name "DEFAULT_VLAN"
   untagged 1-10,13-24,1-24,Trk1
   ipv6 address 2001:db8::214:c2ff:fe4c:e480
exit
spanning-tree Trk1 priority 4
password manager
password operator
```

NOTE: Because the switch's IPv6 address and default gateway were statically configured (not assigned by a DHCP server), when the switch boots up with the IP Preserve startup configuration file (see **Configuring IP preserve**), its current IPv6 address, subnet mask, and default gateway are not changed.

If a switch's current IP address was acquired from a DHCP/Bootp server, the IP Preserve statement is ignored and the IP addresses in the downloaded configuration file are implemented.

For more information on how to use the IP preserve feature, see the current *ArubaOS-Switch Basic Operation Guide.*

This chapter describes management security features that are IPv6 counterparts of IPv4 management security features on the switches. The IPv6-enabled management security features are:

- Authorized IP Managers for IPv6
- Secure Shell 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
- WebAgent
- SNMP (with a correct community name)
- SSH
- TFTP

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.

- You can configure up to 100 authorized IPv4 and IPv6 manager entries on a switch, where each entry applies to either a single management station or a group of stations. Each authorized manager entry consists of an IPv4 or IPv6 address and a mask that determines the individual management stations that are allowed access.
  - You can configure authorized IPv4 manager addresses using the `ip authorized-managers` command. For more information, see "using authorized ip Managers" in the *Access Security Guide*.
  - You can configure authorized IPv6 manager addresses using the `ipv6 authorized-managers` command. For more information, see **Configuring authorized IP managers for switch access** on page 42.
- 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 zeros: **0:0:0:0:0:0:0:0**.)
- You configure each authorized manager address with Manager- or Operator-level privilege to access the switch.

- Manager privilege allows full access to all console interface screens for viewing, configuring, and all other operations available in these interfaces.
- Operator privilege allows read-only access from the 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, 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> ] access-method [all | ssh | telnet | web | snmp | tftp]`

`no ipv6 authorized-managers <ipv6-addr> [ipv6-mask] [access <operator | manager> ] access-method [all | ssh | telnet | web | snmp | tftp]`

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.

Default: `FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF`.

`access <operator|manager>` : Specifies the level of access privilege granted to authorized stations. Applies only to access through Telnet, SSH, and SNMP (version 1, 2, and 3). Default: `Manager`.

`access-method [all | ssh | telnet | web | snmp | tftp]` : Configures access levels by access method and IP address. Each management method can have its own set of authorized managers. Default: `All`.

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

Procedure

1. 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, to authorize only one IPv6-based station for access to the switch.

   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. If you have 10 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.

2. 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). As shown in the table, 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.

Table 1: *Mask for configuring a single authorized IPv6 manager station*

	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	

Multiple station access configuration

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.

If a bit in any of the 4-bit binary representations of a hexadecimal value in a mask is "on" (set to 1), 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.

The table shows the binary expressions represented by individual hexadecimal values in an `ipv6-mask` parameter.

Table 2: *Hexadecimal mask values and binary equivalents*

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

The table below 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`.

Table 3: *Mask for configuring a single authorized IPv6 manager station*

	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 "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	

Table 4: *How a mask determines four authorized IPv6 manager addresses (example)*

Last block in mask: FFFC **Last block in IPv6 address: D37D**																			
Bit numbers	Bit 15	Bit 14		Bit 13	Bit 12	Bit 11	Bit 10		Bit 9	Bit 8	Bit 7	Bit 6		Bit 5	Bit 4	Bit 3	Bit 2		Bit 1	Bit 0
Bit value			F					F					F					C		
FFFC: Last block in mask	1	1		1	1	1	1		1	1	1	1		1	1	1	1		0	0
D37D: Last block in IPv6 address	1	1		0	1	0	0		1	1	0	1		1	1	1	1		0	1
Bit setting:	1 = On				0 = Off															

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 2 bits are set to **0** ("off") and allow the corresponding bits in an authorized IPv6 address to be either "on" or "off".

Table 5: *How hexadecimal C in a mask authorizes four IPv6 manager addresses (example)*

	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	FFFFC
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

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 this block (`FFF8`) of the mask allow for eight possible authorized IPv6 stations:
  2001:DB8:0000:0000:244:17FF:FEB6:D37D2001:DB8:0000:0001:244:17FF:FEB6:D37D
  2001:DB8:0000:0002:244:17FF:FEB6:D37D2001:DB8:0000:0003:244:17FF:FEB6:D37D2001:DB8:0000:0004:244:17FF:FEB6:D37D2001:DB8:0000:0005:244:17FF:FEB6:D37D2001:DB8:0000:0006:244:17FF:FEB6:D37D2001: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.

Table 6: *Mask for configuring Authorized IPv6 Manager stations in different subnets*

	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	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.
IPv6 address	2001	DB8	0000	0000	244	17FF	FEB6	D37D	

The table below 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.

Table 7: *How a mask determines authorized IPv6 manager addresses by subnet*

Fourth block in mask: FFF8 **Fourth Block in Prefix ID of IPv6 Address: 0000**																				
Bit numbers	Bit 15	Bit 14		Bit 13	Bit 12	Bit 11	Bit 10		Bit 9	Bit 8	Bit 7	Bit 6		Bit 5	Bit 4	Bit 3	Bit 2		Bit 1	Bit 0
Bit value			F					F					F					8		
FFFC: Last block in mask	1	1		1	1	1	1		1	1	1	1		1	1	1	0		0	0
D37D:Last block in IPv6 address	0	0		0	0	0	0		0	0	0	0		0	0	0	0		0	0

Bit setting: 1=On; 0=Off.

Viewing an authorized IP managers configuration

Use the `show ipv6 authorized-managers` command to list the IPv6 stations authorized to access the switch.

Output for `show ipv6 authorized-managers`

```
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
```

By analyzing the masks displayed in the table below, the IPv6 stations shown in the table below are granted access.

Table 8: *How masks determine authorized IPv6 manager addresses*

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

Authorizing manager access

The following IPv6 commands authorize manager-level access for one link-local station at a time. 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>`.

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

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

```
Switch(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.

Default IPv6 mask

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

NOTE: 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 next IPv6 command authorizes operator-level access for sixty-four IPv6 stations:32 stations in the subnets defined by 0x0006 and 0x0007 in the fourth block of an authorized IPv6 address:

```
Switch(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:

```
Switch(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 values. 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**:

```
Switch(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). It is not necessary to enter either of these parameters:

```
Switch(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.

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

SCP and SFTP for IPv6

You can take advantage of the SCP and 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.

By default, SSH is enabled for IPv4 and IPv6 connections on a switch, and a single command set is used for both IPv4 and IPv6 file transfers.

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

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

You can perform secure file transfers to and from IPv4 and IPv6 client devices by entering the `ip ssh filetransfer` command.

Syntax:

`ip ssh filetransfer`

`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. See the documentation that comes with an SCP or SFTP client application for information on the file transfer commands and software utilities to use.

NOTE: Enabling SSH file transfer disables TFTP and auto-TFTP operation.

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.

For complete information on how to configure SCP or SFTP in an SSH session to copy files to and from the switch, see the *ArubaOS-Switch Management and Configuration Guide* for your switch.

MLD snooping can be enabled on a VLAN to constrain the flooding of IPv6 multicast traffic on a VLAN. When MLD snooping is enabled, a switch examines MLD messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group.

Multicast addressing

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)

MLD is an IPv6 protocol used on a local link for multicast group management. MLD operates in a manner similar to IGMP in IPv4 networks. MLD is enabled per VLAN and is analogous to the IPv4 IGMP protocol. In the factory default state (MLD disabled), the switch floods all IPv6 multicast traffic it receives on a given VLAN through all ports on that VLAN except the port receiving the inbound multicast traffic. Enabling MLD imposes management controls on IPv6 multicast traffic to reduce unnecessary bandwidth usage. MLD is configured per-VLAN. MLD can be configured using version 1 (MLDv1) or version 2 (MLDv2). MLDv2 introduces source-specific multicast in which the only packets delivered to the receiver are those that originate from a specified source address requested by the receiver. The receiver indicates interest in receiving traffic to a multicast address and additionally can indicate interest in receiving traffic from only one specified source sending to that multicast address. This reduces the amount of multicast routing information that should be maintained.These options are available for MLDv1 and MLDv2:

- Query interval-the time interval between general queries sent by the querier.
- Query Max Response Time-the amount of time to wait for a response to a query.
- Last Member Query Interval-the amount of time 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.
- Robustness-the number of times to retry a query.
- Fast Learn-enables the port to learn group information when there is a topology change.

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.

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 does not interact with it. (However, 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.)

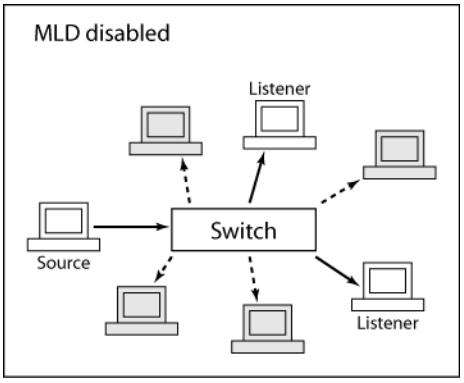
A source node creates multicast traffic by sending packets to a multicast address. In IPv6, addresses with the first 8 bits set (that is, "FF" as the first two characters of the address) are multicast addresses, and any node that listens to such an address will receive the traffic sent to that address. Application software running on the source and destination systems cooperates to determine what multicast address to use. (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, for example, 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.

MLD operation

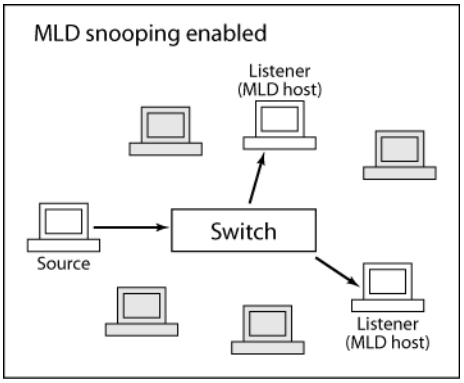
Multicast communication can take place without MLD, and by default, MLD is disabled. In that case, if a switch receives a packet with a multicast destination address, it floods the packet to all ports in the same VLAN (except the port that it came in on), as shown in the following figure. Any network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

Figure 2: *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, as shown in the following figure. It drops that traffic for ports on the VLAN that have no MLD hosts (except for a few special cases explained below).

Figure 3: *With MLD snooping, traffic is sent to MLD hosts*



Forwarding in MLD snooping

When MLD snooping is active, a multicast packet is handled by the switch as shown in the following list.

The packet is:

- Forwarded to ports that have nodes that have joined the packet's multicast address (that is, MLD hosts on that address packet)
- 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 to administratively forwarded ports-The packet is forwarded through all ports set administratively to forward mode. (See the description of forward 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 use "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 prevents IPv6 multicast traffic from moving through the port.

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 (IANA). IPv6 standards define any address beginning with FF0x/12 (binary 1111 1111 0000) as a well-known address.

Enabling or disabling MLD snooping on a VLAN

Several CLI commands are available for configuring MLDv1 and MLDv2 parameters on a switch. To enable or disable MLD on a VLAN, enter the appropriate command.

Syntax

```
ipv6 mld [enable|disable]
```

```
no ipv6 mld [enable|disable]
```

Description

This command enables or disables MLD snooping on a VLAN.

The `no` form disables MLD snooping on a VLAN.

Usage

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

MLDv2 is disabled by default.

enable: Enables MLDv2 on a VLAN.

disable: Disables MLDv2 on a VLAN. The last-saved or the default MLD configuration is saved, whichever is most recent.

Example

This example shows how to enable or disable MLD snooping on a VLAN

To enable MLD snooping on VLAN 500:

```
Switch(vlan-500)# ipv6 mld enable
Switch# config
```

To disable MLD snooping on VLAN 500:

```
Switch(vlan-500)# no ipv6 mld
```

ipv6 mld version

You can specify the MLD version you wish to use with this command.

Syntax

`ipv6 mld version` *`1-2`* `strict`

`no ipv6 mld version` *`1-2`* `strict`

Description

This command sets the MLD protocol version to use.

The `no` version of the command resets the version to the default, version 2.

Usage

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

Example

To set MLD to version 1 for VLAN 8 and version 2 for VLAN 9:

```
Switch(vlan-8)# ipv6 mld version 1
Switch(vlan-8)# exit
Switch(config)# vlan 9
Switch(vlan-9)# ipv6 mld version 2
```

ipv6 mld

Syntax

`ipv6 mld [auto` *`port-list`*`|blocked` *`port-list`*`|forward` *`port-list`*`]`

Description

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

Options

The following settings are allowed:

`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.

port-list

Specifies the affected port or range of ports.

The default value of the filter is `auto`

Example

Configuring per-port MLD traffic filters

```
Switch(vlan-500)# ipv6 mld forward 1-3
Switch(vlan-500)# ipv6 mld blocked 46-47
Switch# show ipv6 mld vlan 500 config

 MLD Service Vlan Config

  VLAN ID                              : 500
  VLAN NAME                            : VLAN500
  MLD Enabled [No]                     : Yes
  Querier Allowed [Yes]                : Yes
  MLD Version                          : 2
  Strict Mode                          : No
  Last Member Query Interval (seconds) : 10
  Robustness-Count                     : 2
  Port Type        Port Mode    Forced Fast Leave  Fast Leave  Fast Learn
  ---- ------------------ ----------------- --------------------------
  1   100/1000T    forward      No                 Yes         No
  2   100/1000T    forward      No                 Yes         No
  3   100/1000T    forward      No                 Yes         No
  46  100/1000T    blocked      No                 Yes         No
  47  100/1000T    blocked      No                 Yes         No
  48  100/1000T    auto         No                 Yes         No

Switch#
```

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 switches 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 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.

ipv6 mld querier

Syntax

`ipv6 mld querier`

`no ipv6 mld querier`

Description

Enables the switch to act 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 defers to that device. If an acting querier stops performing the querier function, all querier-enabled switches and multicast routers on the VLAN enter an election to determine the next device to act as querier.

The `no` form disables the switch from acting as querier on a VLAN.

Usage

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

Example

Configuring the querier

To disable the switch from acting as querier on VLAN 500:

```
Switch(vlan-500)# no ipv6 mld querier
```

To enable the switch to act as querier on VLAN 500:

```
Switch(vlan-500)# ipv6 mld querier
```

ipv6 mld query-interval

Syntax

`ipv6 mld query-interval <60-31744>`

`no ipv6 mld query-interval <60-31744>`

Description

This command specifies the number of seconds between membership queries

The `no` form of the command sets the interval to the default of 125 seconds.

Specifiers

Default: 125 seconds.

Usage

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

Example

To set the query-interval to 300 seconds on ports in VLAN 500:

```
Switch(vlan-500)# ipv6 mld query-interval 300
```

ipv6 mld query-max-response-time

Syntax

`ipv6 mld query-max-response-time <10-128 >`

`no ipv6 mld query-max-response-time <10-128 >`

Description

This command specifies the maximum amount of time to wait for a response to a query

The `no` form of the command sets the interval to the default of 10 seconds.

Usage

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

Example

To set the `query-max-response-time` to 30 seconds on ports on VLAN 500:

```
Switch(vlan-500)# ipv6 mld query-max-response-time 30
```

ipv6 mld robustness

Syntax

`ipv6 mld robustness <1 - 8>`

`no ipv6 mld robustness <1 - 8>`

Description

This command specifies the number of times to retry a query.

The `no` form of the command sets the interval to the default of 2.

Usage

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

Example

```
Switch(vlan-500)# ipv6 mld robustness 4
```

ipv6 mld last-member-query-interval

Syntax

`ipv6 mld last-member-query-interval <1 - 2>`

`no ipv6 mld last-member-query-interval <1 - 2>`

Description

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.

Usage

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

Example

```
switch(vlan-500)# ipv6 mld last-member-query-interval 2
```

ipv6 mld fastlearn

The Fast Learn option allows fast convergence of multicast traffic after a topology change. When a new port joins or moves to a forwarding state, MLD sends joins for the groups it maintains.

For MLDv1, a join is transmitted for each group if the switch is a nonquerier. If the switch is a querier, an MLDv1 query is sent to learn the group on that port.

For MLDv2, an IS_EX report is sent when the switch is a nonquerier. If the switch is a querier, an MLDv2 query is sent on the port to learn the group.

Syntax

`ipv6 mld fastlearn <port-list|all>`

`no ipv6 mld fastlearn <port-list|all>`

Description

This command enables fast learn on the specified ports in a VLAN.

The `no` form of the command disables the fast learn function on the specified ports. The `all` option enabled or disables all ports.

Specifiers

Default: Disabled

Example

```
switch(config)# ipv6 mld fastlearn 5 - 6
```

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 may 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 more complicated, as explained under **Fast leaves and forced fast leaves** on page 58.)

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, 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 several 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 stop (unless the switch sees a new join request).
- If a single port has multiple nodes attached to it, 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 several 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 stop unless the switch sees a new join request. This reduces the number of multicast packets forwarded unnecessarily.

ipv6 mld fastleave

Syntax

`ipv6 mld fastleave` *`port-list`*

`no ipv6 mld fastleave` *`port-list`*

Description

Enables the fast leave function on the specified ports in a VLAN.

The `no` form disables the fast leave function on the specified ports in a VLAN.

Specifiers

Default: Enabled

Usage

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

Example

To disable fast leave on ports in VLAN 8:

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

To enable fast leave on ports in VLAN 8:

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

ipv6 mld forcedfastleave

Syntax

`ipv6 mld forcedfastleave` *`<port-list>`*

`no ipv6 mld forcedfastleave` *`<port-list>`*

Description

Enables the forced fast leave function on the specified ports in a VLAN.

The `no` form disables the forced fast leave function on the specified ports in a VLAN

Specifiers

Default: Disabled

Usage

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

Example

To enable forced fast leave on ports in VLAN 8:

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

To disable forced fast leave on ports in VLAN 8:

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

Current MLD status

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: 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:
  - Multicast group address.
  - 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.
    - MLD version number (MLDv2 display only)
    - Mode-INCLUDE or EXCLUDE (MLDv2 only): when INCLUDE is displayed, the host has requested specific source/group pairs. When EXCLUDE is displayed, the host has requested all sources for a group except for a specified list of sources to exclude.
- Uptime: The length of time the group has been joined.
- Expire time: Time until the group expires if no joins are seen.
- The ports that have joined the multicast group.

show ipv6 mld

Syntax

`show ipv6 mld`

Description

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

Syntax

`show ipv6 mld vlan <vid>`

Description

Displays MLD status for the specified VLAN.

Example

Displaying the MLD configuration for all static VLANs on the switch

```
Switch# show ipv6 mld

  MLD Service Protocol Info
```

```
  Total vlans with MLD enabled               : 1
  Current count of multicast groups joined   : 6

  VLAN ID             : 500
  VLAN Name           : VLAN500
  MLD Version         : 2
  MLD Interface State : Querier
  Querier Address     : fe80::b25a:daff:fe97:6280 [this switch]
          Version     : 2
          Uptime      : 18m 38s
          Expires     : 0m 37s  Querier Up Time : 1h:37m:20s

  Ports with multicast routers :


  Active Group Addresses                Tracking  Vers  Mode  Uptime  Expires
  ------------------------------------ ---- ---------- --------------------
  ff1e::5                               Filtered  2     EXC   3m 7s   2m 57s
  ff1e::6                               Filtered  2     EXC   3m 8s   3m 0s
  ff1e::7                               Filtered  2     EXC   3m 8s   2m 56s
  ff1e::8                               Filtered  2     EXC   3m 8s   2m 57s
  ff1e::9                               Filtered  2     EXC   3m 8s   2m 58s
  ff1e::a                               Filtered  2     EXC   3m 8s   2m 52s

Switch#
```

show ipv6 mld link-local

Syntax

```
show ipv6 mld link-local
```

Description

Displays MLD link-local groups information for all VLANs on the switch that have MLD configured.

Example output

Displaying the MLD link-local configuration for all static VLANs on the switch.

```
switch(config)# show ipv6 mld link-local

  MLD Service Protocol Info

  Total vlans with MLD enabled                : 4
  Current count of multicast groups joined    : 1007

  VLAN ID : 20     NAME : VLAN20
  MLD Version : 2

  MLD Interface State : Non-Querier
  Querier Address     : fe80::3ea8:2aff:fe3c:6c00
          Port        : F22
          Version     : 2
          Uptime      : 13m 3s
          Expires     : 4m 12s
  Ports with multicast routers : F22


  Active Group Addresses                   Tracking Vers Mode Uptime   Expires
  ---------------------------------------- -------- ---- ---- -------- --------
  ff02::1:ff00:50                          Standard 2    EXC  12m 59s  2m 12s
  ff02::1:fff9:e810                        Standard 2    EXC  12m 59s  2m 12s

  VLAN ID : 30     NAME : VLAN30
```

```
MLD Version : 2

MLD Interface State : Non-Querier
Querier Address     : fe80::d6c9:efff:fe96:6b00
        Port        : F23
        Version     : 2
        Uptime      : 12m 14s
        Expires     : 2m 55s
Ports with multicast routers : F23


Active Group Addresses                    Tracking Vers Mode Uptime   Expires
----------------------------------------- -------- ---- ---- -------- --------
ff02::1:ff00:50                           Standard 2    EXC  12m 10s  3m 8s
ff02::1:fffb:685                          Standard 2    EXC  12m 10s  3m 8s

VLAN ID : 40     NAME : VLAN40
MLD Version : 2

MLD Interface State : Querier
Querier Address     : fe80::3664:a9ff:fea5:b00 [this switch]
        Version     : 2
        Uptime      : 11m 24s
        Expires     : 1m 36s
Ports with multicast routers :


Active Group Addresses                    Tracking Vers Mode Uptime   Expires
----------------------------------------- -------- ---- ---- -------- --------
ff02::1:ff00:10                           Standard 2    EXC  11m 18s  3m 55s
ff02::1:fffb:684                          Standard 2    EXC  11m 19s  3m 54s

VLAN ID : 50     NAME : VLAN50
MLD Version : 2

MLD Interface State : Querier
Querier Address     : fe80::3664:a9ff:fea5:b00 [this switch]
        Version     : 2
        Uptime      : 11m 26s
        Expires     : 1m 34s
Ports with multicast routers :


Active Group Addresses                    Tracking Vers Mode Uptime   Expires
----------------------------------------- -------- ---- ---- -------- --------
ff02::5                                   Standard 2    EXC  2m 45s   3m 47s
ff02::6                                   Standard 2    EXC  2m 46s   3m 47s
ff02::7                                   Standard 2    EXC  2m 46s   3m 46s
ff02::8                                   Standard 2    EXC  2m 47s   3m 46s
ff02::9                                   Standard 2    EXC  2m 47s   3m 46s
ff02::a                                   Standard 2    EXC  2m 47s   3m 46s
ff02::b                                   Standard 2    EXC  2m 47s   3m 45s
ff02::c                                   Standard 2    EXC  2m 48s   3m 45s
ff02::d                                   Standard 2    EXC  2m 48s   3m 45s
ff02::e                                   Standard 2    EXC  2m 49s   3m 44s
ff02::f                                   Standard 2    EXC  2m 49s   3m 44s
```

show ipv6 mld vlan link-local

Syntax

`show ipv6 mld vlan <vlan ID> link-local`

Description

Displays MLD link-local groups information for specified VLAN.

Example output

Displaying the MLD link-local configuration for specified VLAN.

```
switch(config)# show ipv6 mld vlan 50 link-local

  MLD Service Protocol Info

  VLAN ID : 50     NAME : VLAN50
  MLD Version : 2

  MLD Interface State : Querier
  Querier Address     : fe80::7210:6fff:fe86:7862 [this switch]
          Version     : 2
          Uptime      : 1h 44m
          Expires     : 0m 15s
  Ports with multicast routers :


  Active Group Addresses                    Tracking Vers Mode Uptime   Expires
  ----------------------------------------- -------- ---- ---- -------- --------
  ff02::5                                   Standard 2    EXC  1h 44m   2m 36s
  ff02::6                                   Standard 2    EXC  1h 44m   2m 35s
  ff02::7                                   Standard 2    EXC  1h 44m   2m 35s
  ff02::8                                   Standard 2    EXC  1h 44m   2m 35s
  ff02::9                                   Standard 2    EXC  1h 44m   2m 35s
  ff02::a                                   Standard 2    EXC  1h 44m   2m 35s
```

Current MLD configuration

The following information applies to all MLD-enabled VLANs:

Control unknown multicast

If this is set to YES, any IPv6 multicast packets that are not joined by an MLD host are 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 are flooded out all ports in the VLAN.

Forced fast leave timeout

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:

- 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 MLD version (1 or 2)

MLD configuration for a specific VLAN

```
Switch# show ipv6 mld vlan 500 config

 MLD Service Vlan Config

  VLAN ID                                 : 500
  VLAN NAME                               : VLAN500
  MLD Enabled [No]                        : Yes
```

```
  Querier Allowed [Yes]                 : Yes
  MLD Version                           : 2
  Strict Mode                           : No
  Last Member Query Interval (seconds)  : 10
  Robustness-Count                      : 2
  Port Type         Port Mode    Forced Fast Leave  Fast Leave  Fast Learn
  ---- ------------------- ---------------- --------------------------
  1    100/1000T    forward      No                 Yes         No
  2    100/1000T    forward      No                 Yes         No
  3    100/1000T    forward      No                 Yes         No
  46   100/1000T    blocked      No                 Yes         No
  47   100/1000T    blocked      No                 Yes         No
  48   100/1000T    auto         No                 Yes         No

Switch#
```

- VLAN ID and Name
- MLD enabled: whether MLD is enabled on the VLAN (default NO, but the information for this VLAN will be listed only if MLD is enabled)
- Querier Allowed: whether the switch is allowed to act as querier on the VLAN
- MLD version
- Strict Mode: whether strict mode is enabled
- Last Member Query Interval: showing the amount of time the querier waits for a response from members, in seconds
- Query Interval showing the length of time between membership queries, in seconds
- Query Max. Response Time displaying the number of seconds to wait for a response to a query, in seconds
- Robustness-Count displaying the number of times to retry a query
- 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 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)
    - `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.
- whether Forced Fast Leave is enabled or disabled

- whether Fast Leave is enabled or disabled
- whether Fast Learn is enabled or disabled - not in sw commands

show ipv6 mld config

Syntax

```
show ipv6 mld config
```

Description

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

Syntax

```
show ipv6 vlan vid [ config ]
```

Description

Displays current MLD configuration for the specified VLAN, including per-port configuration information.

Example

Configuring the current MLD

```
Switch# show ipv6 mld config

 MLD Service Config

  Control unknown multicast [Yes]    : Yes
  Forced fast leave timeout [4]      : 4
  Send Router Alert Option           : Default
  VLAN ID VLAN NAME  MLD Enabled Querier Allowed Version
  ------- ---------- ----------- ----------------------
  500     VLAN500    Yes         Yes              2
```

show ipv6 mld link-local

Syntax

```
show ipv6 mld link-local
```

Description

Displays MLD link-local groups information for all VLANs on the switch that have MLD configured.

Example output

Displaying the MLD link-local configuration for all static VLANs on the switch.

```
switch(config)# show ipv6 mld link-local

  MLD Service Protocol Info

  Total vlans with MLD enabled                : 4
  Current count of multicast groups joined    : 1007

  VLAN ID : 20     NAME : VLAN20
  MLD Version : 2

  MLD Interface State : Non-Querier
  Querier Address     : fe80::3ea8:2aff:fe3c:6c00
          Port        : F22
          Version     : 2
          Uptime      : 13m 3s
```

```
        Expires     : 4m 12s
Ports with multicast routers : F22


Active Group Addresses                    Tracking Vers Mode Uptime   Expires
----------------------------------------- -------- ---- ---- -------- --------
ff02::1:ff00:50                           Standard 2    EXC  12m 59s  2m 12s
ff02::1:fff9:e810                         Standard 2    EXC  12m 59s  2m 12s

VLAN ID : 30     NAME : VLAN30
MLD Version : 2

MLD Interface State : Non-Querier
Querier Address     : fe80::d6c9:efff:fe96:6b00
        Port        : F23
        Version     : 2
        Uptime      : 12m 14s
        Expires     : 2m 55s
Ports with multicast routers : F23


Active Group Addresses                    Tracking Vers Mode Uptime   Expires
----------------------------------------- -------- ---- ---- -------- --------
ff02::1:ff00:50                           Standard 2    EXC  12m 10s  3m 8s
ff02::1:fffb:685                          Standard 2    EXC  12m 10s  3m 8s

VLAN ID : 40     NAME : VLAN40
MLD Version : 2

MLD Interface State : Querier
Querier Address     : fe80::3664:a9ff:fea5:b00 [this switch]
        Version     : 2
        Uptime      : 11m 24s
        Expires     : 1m 36s
Ports with multicast routers :


Active Group Addresses                    Tracking Vers Mode Uptime   Expires
----------------------------------------- -------- ---- ---- -------- --------
ff02::1:ff00:10                           Standard 2    EXC  11m 18s  3m 55s
ff02::1:fffb:684                          Standard 2    EXC  11m 19s  3m 54s

VLAN ID : 50     NAME : VLAN50
MLD Version : 2

MLD Interface State : Querier
Querier Address     : fe80::3664:a9ff:fea5:b00 [this switch]
        Version     : 2
        Uptime      : 11m 26s
        Expires     : 1m 34s
Ports with multicast routers :


Active Group Addresses                    Tracking Vers Mode Uptime   Expires
----------------------------------------- -------- ---- ---- -------- --------
ff02::5                                   Standard 2    EXC  2m 45s   3m 47s
ff02::6                                   Standard 2    EXC  2m 46s   3m 47s
ff02::7                                   Standard 2    EXC  2m 46s   3m 46s
ff02::8                                   Standard 2    EXC  2m 47s   3m 46s
ff02::9                                   Standard 2    EXC  2m 47s   3m 46s
ff02::a                                   Standard 2    EXC  2m 47s   3m 46s
ff02::b                                   Standard 2    EXC  2m 47s   3m 45s
ff02::c                                   Standard 2    EXC  2m 48s   3m 45s
ff02::d                                   Standard 2    EXC  2m 48s   3m 45s
```

```
  ff02::e                                  Standard 2    EXC  2m 49s   3m 44s
  ff02::f                                  Standard 2    EXC  2m 49s   3m 44s
```

show ipv6 mld vlan link-local

Syntax

```
show ipv6 mld vlan <vlan ID> link-local
```

Description

Displays MLD link-local groups information for specified VLAN.

Example output

Displaying the MLD link-local configuration for specified VLAN.

```
switch(config)# show ipv6 mld vlan 50 link-local

  MLD Service Protocol Info

  VLAN ID : 50     NAME : VLAN50
  MLD Version : 2

  MLD Interface State : Querier
  Querier Address     : fe80::7210:6fff:fe86:7862 [this switch]
          Version     : 2
          Uptime      : 1h 44m
          Expires     : 0m 15s
  Ports with multicast routers :


  Active Group Addresses                   Tracking Vers Mode Uptime   Expires
  ---------------------------------------- -------- ---- ---- -------- --------
  ff02::5                                  Standard 2    EXC  1h 44m   2m 36s
  ff02::6                                  Standard 2    EXC  1h 44m   2m 35s
  ff02::7                                  Standard 2    EXC  1h 44m   2m 35s
  ff02::8                                  Standard 2    EXC  1h 44m   2m 35s
  ff02::9                                  Standard 2    EXC  1h 44m   2m 35s
  ff02::a                                  Standard 2    EXC  1h 44m   2m 35s
```

Commands to list currently joined ports

Syntax

```
show ipv6 mld vlan vid group
```

Description

Lists the ports currently joined for all IPv6 multicast group addresses in the specified VLAN.

Syntax

```
show ipv6 mld vlan vid group ipv6-addr
```

Description

Lists the ports currently joined for the specified IPv6 multicast group address in the specified VLAN.

Syntax

```
show ipv6 mld vlan vid group port-num
```

Description

Shows a list of all the MLD groups on the specified port.

Syntax

`show ipv6 mld vlan vid group ipv6-addr source ipv6-addr`

Description

Only for MLDv2. Specify the source IPv6 address.

Ports Joined to multicast groups in a specific VLAN

```
Switch# show ipv6 mld vlan 500 group

 MDL Service Protocol Group Info

 VLAN ID   : 500
 VLAN Name : VLAN500

 Group Address : ff1e::5
 Last Reporter : fe80::200:1ff:fe1b:c3a1
 Group Type    : Filtered

 Port Vers Mode Uptime   Expires  V1     Filter  Sources    Sources
                                 Timer  Timer   Forwarded  Blocked
 ---- ------------------------------------------------------------
 2    2    EXC  32m 18s  2m 59s   -      2m 59s  0          0

 Group Address : ff1e::6
 Last Reporter : fe80::200:1ff:fe1b:c3a1
 Group Type    : Filtered

 Port Vers Mode Uptime   Expires  V1     Filter  Sources    Sources
                                 Timer  Timer   Forwarded  Blocked
 ---- ------------------------------------------------------------
 2    2    EXC  32m 19s   2m 56s   -     2m 56s  0          0

 Port Vers Mode Uptime   Expires  V1     Filter  Sources    Sources
                                 Timer  Timer   Forwarded  Blocked
 ---- ------------------------------------------------------------
 1    2    INC  0m 19s  4m 13s   -      -       1          0

 Group Address  : ff1e::7
 Source Address : 5555::1
 Source Type    : Filtered

 Port Mode Uptime   Expires  Configured Mode
 ---- ---------------------------------
 1    INC  0m 19s   4m 13s   Auto

 Group Address : ff1e::8
 Last Reporter : fe80::200:1ff:fe1b:c3a1
 Group Type    : Filtered

 Port Vers Mode Uptime   Expires  V1     Filter  Sources    Sources
                                 Timer  Timer   Forwarded  Blocked
 ---- ------------------------------------------------------------
 1    2    INC  0m 24s  4m 15s   -      -       1          0
```

show ipv6 mld statistics

Syntax

`show ipv6 mld statistics`

`show ipv6 mld vlan vid statistics`

Description

The general form 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.

MLD statistics for all VLANs configured

```
Switch# show ipv6 mld 500 statistics

 MLD Statistics
  VLAN ID   : 500
  VLAN NAME : VLAN500
  Number of Filtered Groups             : 6
  Number of Standard groups             : 0
  Total Multicast Groups Joined         : 6


 Mode      EXCLUDE    INCLUDE
 ------- ------------ ------
 Filtered  6          1
 Standard  0          0
 Total     6          0
```

Counters

The following information is shown:

- VLAN number and name
- For each VLAN, number of:
  - general queries(MLDv1) received and sent
  - general queries (MLDv2) received and sent
  - version 1 group-specific queries received and sent
  - version 2 group-specific queries received and sent
  - group and source-specific queries received and sent
  - MLD version2 member reports (joins) received
  - MLD version 1 member reports (joins) received
  - version 1 leaves received and sent
  - packets forwarded to routers on this VLAN received and sent
  - packets forwarded to all ports on this VLAN received and sent
- Errors, number of:
  - MLD packets of unknown type received
  - malformed packets received
  - packets with bad checksums
  - packets from a martian source (the wrong subnet on an interface)

- packets received on an MLD-disabled interface
- queries—when a VLAN is configured as MLDv2 and an MLDv1 query is received from another switch for that VLAN, this counter is incremented. The reverse also applies.

- Port Counters, number of:
  - fast leaves that have occurred
  - forced fast leaves that have occurred
  - times a join has timed out on this VLAN

show ipv6 mld vlan counters

Syntax

`show ipv6 mld vlan vid counters`

Description

Displays MLD counters for the specified VLAN.

MLD counters for a single VLAN

```
Switch# show ipv6 mld vlan 500 counters

 MLD Service Vlan Counters

  VLAN ID   : 500
  VLAN NAME : VLAN500


                                          Rx   Tx
                                          -------
 V1 All Hosts Query                       0    2
 V2 All Hosts Query                       0    24
 V1 Group Specific Query                  0    0
 V2 Group Specific Query                  0    12
 Group and Source Specific Query          0    0
 V2 Member Report                         220  0
 V1 Member Join                           12   0
 V1 Member Leave                          6    0
 Forward to Routers                       0    238
 Forward to VLAN                          0    26

 Errors:

 Unknown MLD Type                         0
 Unknown Packet                           0
 Malformed Packet                         12
 Bad Checksum                             0
 Martian Source                           0
 Packet received on MLD-disabled Interface  0
 Interface Wrong Version Query            0

 Port Counters:

 Fast Leave                               0
 Forced Fast Leave                        0
 Membership Timeout                       0

Switch#
```

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 malformed packets
  - number of packets received on MLD-disabled Interface
  - 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

Reset MLD state

The **`mld reload`** command resets MLD state on all interfaces.

MLD reset

```
Switch(vlan-500)# show ipv6 mld
       MLD application is in Error State as System Resources are exhausted.
       Traffic will flood. Disable MLD on all VLANs or Issue the Command
       "mld reload" to take it out of Error. Refer to your product manual
       for information on MLD resource consumption.
```

Router alert

Syntax

```
ipv6 mld send-router-alert default
ipv6 mld send-router-alert alternative-padding
```

Parameters

`send-router-alert`

Enables/Disables insertion of the Router Alert option.

`default`

Enable insertion of the Router Alert option

`alternative-padding`

Enable insertion of the Router Alert option with padding in between the header length and the option type for interoperability.

Listeners and joins

The "snooping" part of MLD snooping arises because a switch must track which ports have network nodes that are MLD hosts for any given multicast address. It does this by tracking "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.

Introduction to IPv6 ACLs

An Access Control List (ACL) contains one or more Access Control Entries (ACEs) specifying the criteria the switch uses to either permit (forward) or deny (drop) IP packets traversing the switch's interfaces.

This chapter describes how to configure, apply, and edit static IPv6 ACLs for filtering IPv6 traffic in a network populated with the switches covered by this guide, and how to monitor IPv6 ACL actions.

NOTE: Because the switches covered by this guide operate in an IPv4/IPv6 dual stack mode, IPv6 and IPv4 ACLs can operate simultaneously in these switches. However:

- Static IPv6 ACLs and IPv4 ACLs do not filter each other's traffic.
- IPv6 and IPv4 ACEs cannot be configured in the same static ACL.
- RADIUS-assigned ACLs can be configured to filter either IPv4 traffic only, or both IPv4 and IPv6 traffic.

In this chapter, unless otherwise noted:

- The term "ACL" refers to IPv6 ACLs.
- Descriptions of ACL operation apply only to IPv6 traffic.

For information on configuring and applying static IPv4 ACLs, see the chapter titled "IPv4 Access Control Lists (ACLs)" in the *ArubaOS-Switch Access Security Guide* for your switch.

IPv6 traffic filtering with ACLs can help to improve network performance and restrict network use by creating policies for:

- Switch Management Access: Permits or denies in-band management access. This includes limiting and/or preventing the use of designated protocols that run on top of IPv6, such as TCP, UDP, ICMP, and others. Also included are the use of DSCP criteria, and control for application transactions based on source and destination IPv6 addresses and transport layer port numbers.
- Application Access Security: Eliminates unwanted IPv6 traffic in a path by filtering IPv6 packets where they enter or leave the switch on specific VLAN interfaces.

CAUTION: The ACLs described in this chapter can enhance network security by blocking selected IPv6 traffic, and can serve as part of your network security program. However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IPv6 packet transmissions, they should not be relied upon for a complete security solution.

Static IPv6 ACLs on the switches covered by this manual do not screen non- IPv6 traffic such as IPv4, AppleTalk, and IPX packets.

Overview of IPv6 ACLs

The IPv6 ACLs overview contains types of ACLs, concurrent IPv4 and IPv6 ACLs, ACL inbound application points, VACL applications, RADIUS-assigned (dynamic) port ACL applications, 802.1X user-based and port-based applications, and operating notes for IPv6 applications.

Types of ACLs

A permit or deny policy for IPv6 traffic you want to filter is based on source and destination IPv6 address, plus other IPv6 protocol factors such as TCP/ UDP, ICMP, and DSCP.

802.1X user-based and port-based applications

User-Based 802.1X access control allows up to 32 individually authenticated clients on a given port. Port-Based access control does not set a client limit and requires only one authenticated client to open a given port (and is recommended for applications where only one client at a time can connect to the port).

- If you configure 802.1X user-based security on a port and the RADIUS response includes a RADIUS-assigned ACL for at least one authenticated client, the RADIUS response for all other clients authenticated on the ports must also include a RADIUS-assigned ACL. Inbound IP traffic on the port from a client that authenticates without receiving a RADIUS-assigned ACL is dropped and the client de-authenticated.
- Using 802.1X port-based security on a port where the RADIUS response to a client authenticating includes a RADIUS-assigned ACL, different results can occur, depending on whether any additional clients attempt to use the port and whether these other clients initiate an authentication attempt. This option is recommended for applications where only one client at a time can connect to the port, and not recommended for instances where multiple clients may access the same port at the same time. For more information, see the latest *ArubaOS-Switch Access Security Guide* for your switch.

Operating notes for IPv6 applications

- For RADIUS ACL applications the switch operates in a dual-stack mode, and a RADIUS-assigned ACL filters both IPv4 and IPv6 traffic. At a minimum, a RADIUS-assigned ACL automatically includes the implicit deny for both IPv4 and IPv6 traffic. Thus, an ACL configured on a RADIUS server to filter IPv4 traffic also denies inbound IPv6 traffic from an authenticated client unless the ACL includes ACEs that permit the desired IPv6 traffic. The reverse is true for a dynamic ACL configured on RADIUS server to filter IPv6 traffic. (ACLs are based on the MAC address of the authenticating client.) For more information, see the latest *ArubaOS-Switch Access Security Guide* for your switch.
- To support authentication of IPv6 clients:
  - The VLAN to which the port belongs must be configured with an IPv6 address.
  - Connection to an IPv6-capable RADIUS server must be supported.
- For 802.1X or MAC authentication methods, clients can authenticate regardless of their IP version (IPv4 or IPv6).
- For the web authentication method, clients must authenticate using IPv4. However, this does not prevent the client from using a dual stack, or the port receiving a RADIUS-assigned ACL configured with ACEs to filter IPv6 traffic.
- The RADIUS server must support IPv4 and have an IPv4 address. RADIUS clients can be dual stack, IPv6-only, or IPv4-only.
- 802.1X rules for client access apply to both IPv6 and IPv4 clients for RADIUS-assigned ACLs. See **802.1X user-based and port-based applications** on page 74.

Features common to all ACLs

- Any ACL can have multiple entries (ACEs).
- You can apply any one ACL to multiple interfaces.

- All ACEs in an ACL configured on the switch are automatically sequenced (numbered). For an existing ACL, entering an ACE without specifying a sequence number automatically places the ACE at the end of the list. Specifying a sequence number inserts the ACE into the list at the specified sequential location.
  - Automatic sequence numbering begins with “10” and increases in increments of 10. You can renumber the ACEs in an ACL and also change the sequence increment between ACEs.
  - The CLI `remark` command option allows you to enter a separate comment for each ACE.
- A source or destination IP address and a mask, together, can define a single host, a range of hosts, or all hosts.
- Every ACL populated with one or more explicit ACEs automatically includes an Implicit Deny as the last entry in the list. The switch applies this action to packets that do not match other criteria in the ACL.
- In any ACL, you can apply an ACL log function to ACEs that have an explicit “deny” action. (The logging occurs when there is a match on a “deny” ACE that includes the `log` keyword.) The switch sends ACL logging output to Syslog, if configured, and optionally, to a console session.

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. See **Creating or editing ACLs offline** on page 111.

IPv6 ACL operation

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 interfaces, and offer these traffic filtering options:

- IPv6 traffic inbound/outbound on a port.
- IPv6 traffic inbound/outbound on a VLAN.
- Routed IPv6 traffic entering or leaving the switch on a VLAN. (Note that ACLs do not screen traffic at the internal point where traffic moves between VLANs or subnets within the switch).

The following table lists the range of interface options:

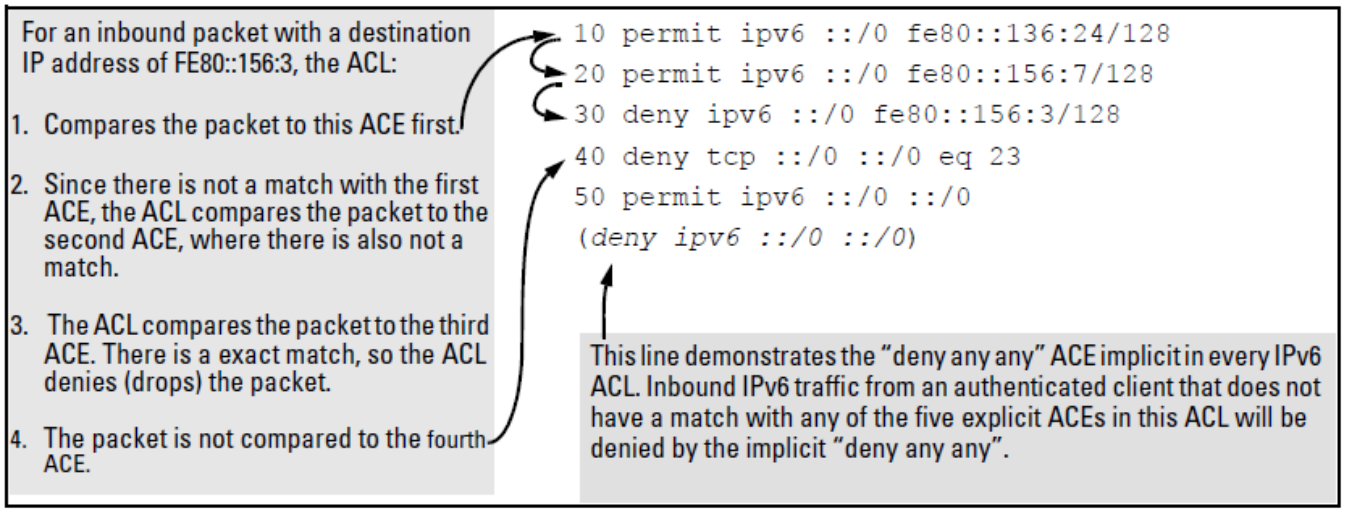
Interface	ACL Application	Application Point	Filter Action
Port	Static Port ACL (switch configured) RADIUS-assigned ACL1	inbound/outbound on the switch port inbound/ outbound on the switch port used by authenticated client	inbound/outbound IPv6 traffic inbound/outbound IPv6 traffic from the authenticated client
VLAN	VACL	entering or leaving the switch on the VLAN	inbound or outbound IPv6 traffic

NOTE: After you assign an ACL to an interface, the default action on the interface is to implicitly deny any IPv6 traffic that is not specifically permitted by the ACL. (This applies only in the direction of traffic flow filtered by the ACL.)

The packet-filtering process

Sequential comparison and action: When an ACL filters a packet, it sequentially compares each ACE's filtering criteria to the corresponding data in the packet until it finds a match. The action indicated by the matching ACE (deny or permit) is then performed on the packet.

Figure 4: *Example of sequential comparison*

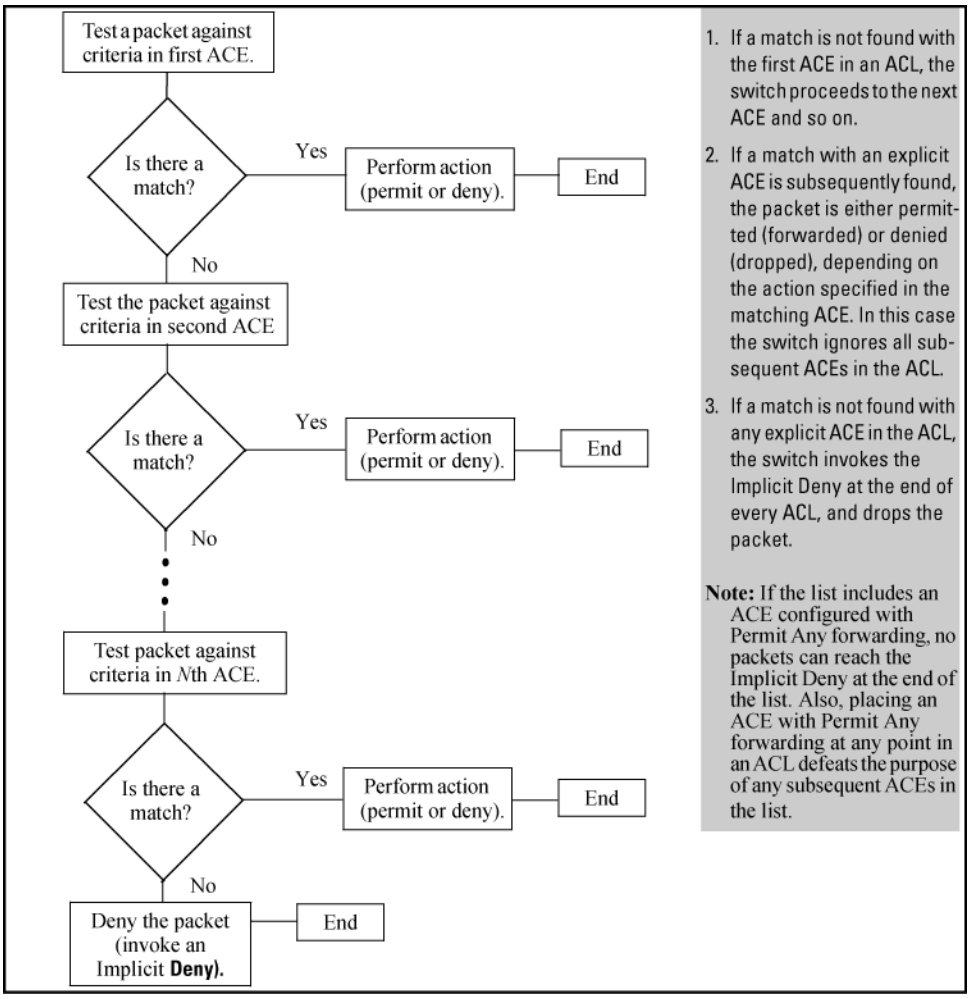


As shown above, the ACL tries to apply the first ACE in the list. If there is not a match, it tries the second ACE, and so on. When a match is found, the ACL invokes the configured action for that entry (permit or drop the packet) and no further comparisons of the packet are made with the remaining ACEs 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 ACL denies (drops) the packet. If you need to override the implicit deny so that a packet that does not have a match will be permitted, then configure `permit ipv6 any any` as the last ACE in the ACL. This directs the ACL to permit

(forward) 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 ipv6 any any`.

Figure 5: *Packet-filtering process in an ACL with N entries (ACEs)*



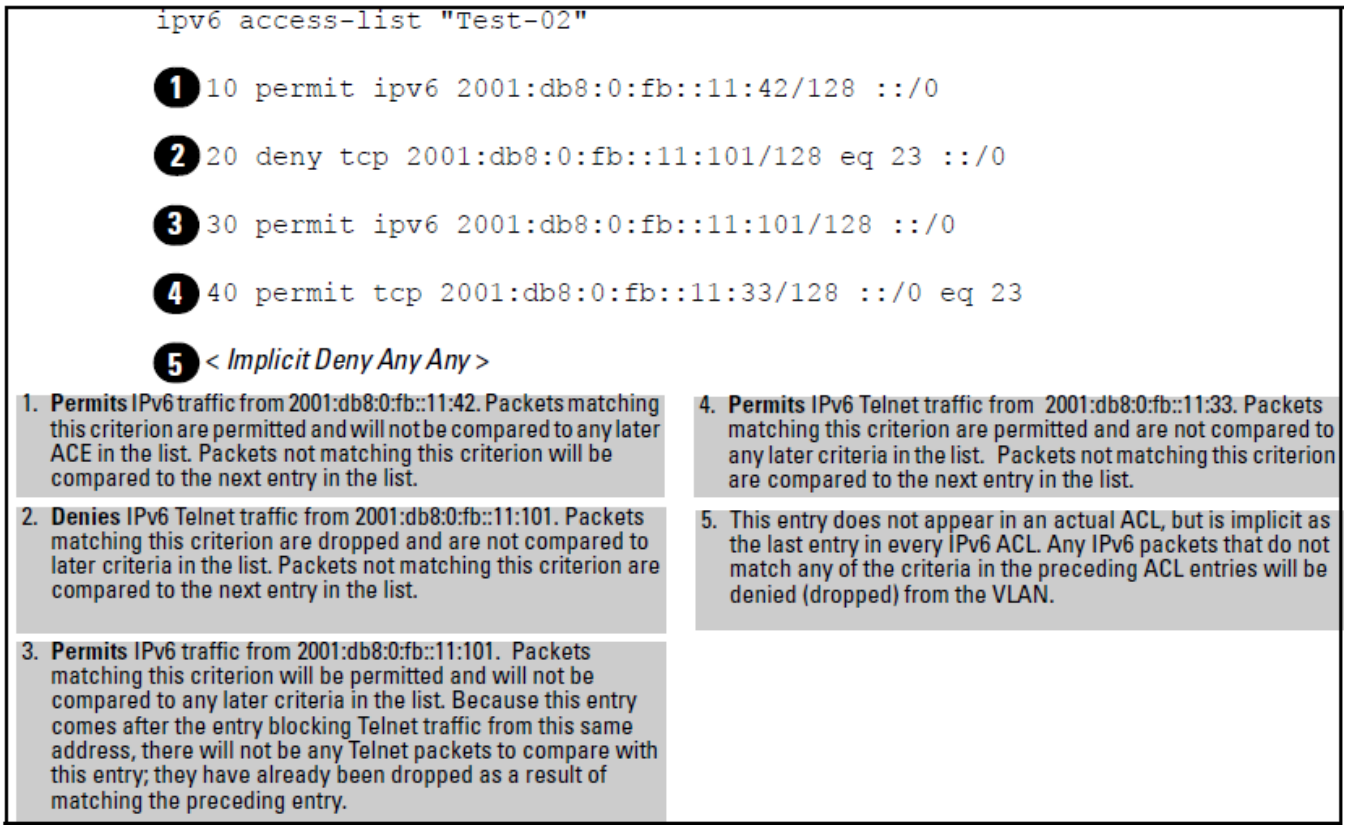
NOTE: The order in which an ACE occurs in an ACL is significant. For example, if an ACL contains six ACEs, but the first ACE allows “Permit Any” forwarding, then the ACL permits all IPv6 traffic, and the remaining ACEs in the list do not apply, even if they have a match with any traffic permitted by 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 6: *Example of how an ACL filters packets*



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

```
Switch(config)# vlan 100 ipv6 access-group Test-02 <vlan-in|vlan-out>
```

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:

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

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.)

IPv6 traffic management and improved network performance

You can use ACLs to block IPv6 traffic from individual hosts, workgroups, or subnets, and to block access to VLANs, subnets, devices, and services. Traffic criteria for ACLs include:

- Switched IPv6 traffic
- Switched and/or routed IPv6 traffic

- IPv6 traffic of a specific protocol type (0-255)
- TCP traffic (only) for a specific TCP port or range of ports, including optional control of connection traffic based on whether the initial request should be allowed
- UDP traffic (only) or UDP traffic for a specific UDP port
- ICMP traffic (only) or ICMP traffic of a specific type and code
- Any of the above with specific precedence and/or ToS settings

Depending on the source and/or destination of a given IPv6 traffic type, you must also determine the ACL application(s) (VACL or static port ACL) needed to filter the traffic on the applicable switch interfaces. Depending on the source and/or destination of a given IPv6 traffic type, you must also determine the ACL application(s) (RACL, VACL, or static port ACL) needed to filter the traffic on the applicable switch interfaces. 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 IPv6 traffic, and what ACL application(s) should be used? In many cases it makes sense to prevent unwanted IPv6 traffic from reaching the core of your network by configuring ACLs to drop unwanted IPv6 traffic at or close to the edge of the network. (The earlier in the network path you can deny unwanted traffic, the greater the benefit for network performance.)
- From where is the traffic coming? The source and destination of IPv6 traffic you want to filter determines the ACL application to use (VACL, static port ACL, and RADIUS-assigned ACL). The source and destination of IPv6 traffic you want to filter determines the ACL application to use (RACL, VACL, static port ACL, and RADIUS-assigned ACL).
- What IPv6 traffic should you explicitly deny? 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.
- What IPv6 traffic can you implicitly deny by taking advantage of the implicit `deny ipv6 any any` to deny IPv6 traffic that you have not explicitly permitted? This can reduce the number of entries needed in an ACL.
- What IPv6 traffic should you permit? In some cases you will need to explicitly identify permitted IPv6 traffic. In other cases, depending on your policies, you can insert an ACE with “permit any” forwarding at the end of an ACL. This means that IPv6 traffic not specifically matched by earlier entries in the list will be permitted.

Security

ACLs can enhance security by blocking IPv6 traffic carrying an unauthorized source IPv6 address (SA). This can include:

- blocking access from specific devices or interfaces (port or VLAN)
- 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 specific TCP, UDP, and ICMP traffic types, including unauthorized access using functions such as Telnet, SSH, and the WebAgent

You can also enhance switch management security by using ACLs to block IPv6 traffic that has the switch itself as the destination address (DA).

NOTE: 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.

Guidelines for planning the structure of an ACL

After determining the ACL application (VACL or static port ACL) to use at a particular point in your network, determine the order in which to apply individual ACEs to filter IPv6 traffic. After determining the ACL application (RACL, VACL, or static port ACL) to use at a particular point in your network, determine the order in which to apply individual ACEs to filter IPv6 traffic.

- The sequence of ACEs is significant. When the switch uses an ACL to determine whether to permit or deny a packet on a particular VLAN, 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.
- The first match in an ACL dictates the action on a packet. Subsequent matches in the same ACL are ignored. However, if a packet is permitted by one ACL assigned to an interface, but denied by another ACL assigned to the same interface, the packet will be denied on the interface.
- On any ACL, the switch implicitly denies IPv6 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, append an ACE that enables Permit Any forwarding as the last ACE in an ACL. This ensures that no packets reach the Implicit Deny case for that ACL.
- Generally, you should list ACEs from the most specific (individual hosts) to the most general (subnets or groups of subnets) unless doing so permits IPv6 traffic that you want dropped. For example, an ACE allowing a series 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

- RACLs and routed IPv6 traffic: Except for IPv6 traffic with a DA on the switch itself, RACLs filter only routed IPv6 traffic that is entering or leaving the switch on a given VLAN. Thus, if routing is not enabled on the switch, there is no routed IPv6 traffic for RACLs to filter.

  VACLs and switched or routed IPv6 traffic: A VACL filters IPv6 traffic leaving the switch on the VLAN(s) to which it is assigned.
- Per switch ACL limits for all ACL types: At a minimum an ACL must have one, explicit “permit” or “deny” Access Control Entry. You can configure up to 2048 ACLs (IPv4 and IPv6 combined). Total ACEs in all ACLs depends on the combined resource usage by ACL and other features.
- Implicit deny: In any static ACL, the switch implicitly (automatically) applies an implicit `deny ipv6 any 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 IPv6 packets that you have not expressly denied, you must enter a `permit ipv6 any any` as the last 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 a `permit ipv6 any any` entry will be permitted, and will not encounter the implicit “Deny” ACE the switch automatically includes at the end of the ACL (see **An ACE that permits all IPv6 traffic not implicitly denied** on page 86). For implicit deny operation in RADIUS-assigned (dynamic) ACLs, see “Configuring RADIUS Server Support for Switch Services” in the latest *Access Security Guide* for your Switch.
- Explicitly permitting IPv6 traffic: Entering a `permit ipv6 any any` ACE in an ACL permits the IPv6 traffic not previously permitted or denied by that ACL. Any ACEs listed after that point do not have any effect.

- Explicitly denying IPv6 traffic: Entering a `deny ipv6 any any` ACE in an ACL denies IPv6 traffic not previously permitted or denied by that ACL. Any ACEs listed after that point have no effect.
- Replacing one ACL with another of the same type: For a specific interface, the most recent ACL assignment using a given application replaces any previous ACL assignment using the same application on the same interface. For example, if you assigned a VACL named “Test-01” to filter inbound or outbound IPv6 traffic on VLAN 20, butlater, you assigned another VACL named “Test-02” to filter inbound IPv6 traffic on this same VLAN, VACL “Test-02” replaces VACL “Test- 01” as the ACL to use. For example, if you assigned an RACL named “Test-01” to filter inbound routed IPv6 traffic on VLAN 20, but later, you assigned another RACL named “Test-02” to filter inbound routed IPv6 traffic on this same VLAN, RACL “Test-02” replaces RACL “Test- 01” as the ACL to use.
- Static port ACLs: These are applied per-port, per port-list, or per static trunk. Adding a port to a trunk applies the trunk’s ACL configuration to the new member. If a port is configured with an ACL, the ACL must be removed before the port is added to the trunk. Also, removing a port from an ACL-configured trunk removes the ACL configuration from that port.
- VACLs: These filter IPv6 traffic leaving the switch through any port belonging to the designated VLAN.
- VACLs operate on static VLANs: You can assign an ACL to any VLAN that is statically configured on the switch. ACLs do not operate with dynamic VLANs.
- VACLs and RACLs operate on static VLANs: You can assign an ACL to any VLAN that is statically configured on the switch. ACLs do not operate with dynamic VLANs.
- A VACL affects all physical ports in a static VLAN: A VACL assigned to a VLAN applies to all physical ports on the switch belonging to that VLAN, including ports that have dynamically joined the VLAN.
- RACLs screen routed IPv6 traffic entering or leaving the switch on a given VLAN interface: This means that the following traffic is subject to ACL filtering:

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::/482001: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.

SA/DA prefix lengths

```
permit tcp 2001:db8:10:1::/64 eq 23 2001:db8:10:1:218:71ff:fec4::/112
```

`permit tcp 2001:db8:10:1::/64`: The prefix defining the mask for the leading bits in the source address.

`eq 23 2001:db8:10:1:218:71ff:fec4::/112`: The prefix defining the mask for the leading bits in the destination address.

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>* :0to*<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.	2530:0:a03:e102:215:60ff:fe7a:adc0 /128	All bits. Used for a specific SA or DA.
	2530:0: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.
	2530:0:a03:e102:215:60ff:fe7a:adc0 /64	Global unicast address with a prefix of 64 bits and a device ID of 64 bits.

Table Continued

Prefix usage	Examples	Notes
For a router advertisement (RA), the included prefix defines the network or range of networks, and the subnets the router is advertising.	2530:0:a03::/48	An RA with a 48-Bit Prefix
	2530:0:a03:e102::/64	An RA with a 64-Bit Prefix

Planning an ACL application

Before creating and implementing ACLs, define the policies you want your ACLs to enforce, and understand how the ACL assignments will impact your network users.

NOTE: IPv6 traffic leaving the switch on a given interface is filtered by the ACLs configured for inbound traffic on that interface. For this reason, an inbound packet will be denied (dropped) if it has a match with an implicit (or explicit) `deny ipv6 any any` in any of the inbound ACLs applied to the interface.

Configuring and assigning an ACL

Overview of configuring and assigning an ACL

This section gives an overview of configuring and assigning an ACL.

General steps for implementing ACLs

1. Configure one or more ACLs. This creates and stores the ACL(s) in the switch configuration.
2. Assign an ACL. This step uses one of the following applications to assign the ACL to an interface:
   - VACL (IPv6 traffic entering or leaving the switch on a given VLAN)
   - Static Port ACL (IPv6 traffic entering or leaving the switch on a given port, port list, or static trunk)

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.

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)

General structure options for an IPv6 ACL

```
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]
   . . .
    <Implicit Deny Any Any>
   exit
```

The ACL in the example below filters traffic for individual hosts in some instances and all hosts in others:

Displayed ACL configuration

```
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
```

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.

Table Continued

Line	Action
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.

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 has the same sequence number as the ACE. (One remark is allowed per ACE.) See **Attaching a remark to an ACE** on page 102.
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 a VLAN or port depends on the concurrent resource usage by multiple configured features. For more information, use the `show <qos|access-list> resources` command.

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.)

ACL configuration factors

This section describes various factors responsible for configuring an ACL.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. If you configure the switch to use an ACL for filtering either inbound or outbound traffic on a VLAN, any IPv6 packet not specifically permitted or denied by the explicit entries you create is denied by the implicit deny action.

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.

Suppose that you have applied the ACL shown in the example below, to inbound IPv6 traffic on VLAN 1 (the default VLAN):

An ACE that permits all IPv6 traffic not implicitly denied

```
ipv6 access-list "Sample-List-2"
  10 deny ipv6 2001:db8::235:10 /128 ::/0
  20 deny ipv6 2001:db8::245:89/128 ::/0
  30 permit tcp 2001:db8::18:100/128 2001:db8::237:1/128
  40 deny tcp 2001:db8::18:100/128 ::/0
  50 permit ipv6 ::/0 ::/0
  (Implicit deny ipv6 any any)
  exit
```

Note the following: Source Address, Prefix length and Destination Address (specifies any IPv6 destination).

Note: After the last explicit ACE there is always an Implicit Deny. However, in this case it will not be used because the last `permit ipv6` ACL allows all IPv6 packets that earlier ACEs have not already permitted or denied.

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.)

Table Continued

Line #	Action
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 is always 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; see **An ACE that permits all IPv6 traffic not implicitly denied** on page 86. This means that if you configure the switch to use an ACL for filtering either inbound or outbound traffic on a VLAN, any IPv6 packets not specifically permitted or denied by the explicit entries you create is 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 is permitted), insert an explicit `permit ipv6 any any` as the last explicit ACE in the ACL.

A configured ACL has no effect until applied 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.) See the latest version of the *ArubaOS-Switch Management and Configuration Guide* for your switch.

Assignment of an ACL name to an interface

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. 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.

(RADIUS-based ACL resources are drawn from the IPv4 allocation).

Creating an ACL using the CLI

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.

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:

Procedure

1. 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.)
2. Enter the text of the ACE without specifying a sequence number.
3. 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.
   ```
   Switch(config)# ipv6 access-list List-1 Switch(config-ipv6-acl)# permit host
   2001:db8:0:a9::8d:100 any
   ```
4. To insert an ACE anywhere in an existing ACL:
5. Enter the context of the ACL and specify a sequence number.
6. 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:
   ```
   Switch(config)# ipv6 access-list List-2 Switch(config-ipv6-acl)# deny host
   2001:db8:0:a9::8d:77 any
   ```

Deleting an ACE

Procedure

1. Enter the **ACL** context and delete the sequence number for the unwanted ACE.
2. 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", enter the following commands:

   ```
   Switch(config)# ipv6 access-list List-2 config
   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, the addresses match.

Table 9: *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
2530:0:a03:e102::/64	2530:0:a03:e102	The leftmost 64 bits must match. The remaining 64 bits are wildcards.
2530:0:a03:e102:215::/80	2530:0:a03:e102:215	The leftmost 80 bits must match. The remaining 48 bits are wildcards.
2530:0:a03:e102:215:60ff:fe7a:adc0/128	2530:0: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

This section describes the commands used to create, enter, and configure an ACL.

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, as well as any IPv6 protocol-specific criteria included in the command.

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 and consists of an alphanumeric string of up to 64 case-sensitive characters. If you include spaces in the string, you must enclose the string in single or double quotes. For example: "Accounting ACL".

You can also use this command to access an existing ACL; see **General editing rules** on page 98.

Entering the ACL context

```
Switch(config)# ip access-list Sample-List
Switch(config-ipv6-acl)#
```

Configuring ACEs in an ACL

Configuring ACEs is done after using the `ipv6 access-list <ascii-str>` command to enter the IPv6 ACL (**ipv6-acl**) context of an ACL.

Syntax:

`<deny|permit> <ipv6|ipv6-protocol|ipv6-protocol-nbr>`

`<any|host <SA>|SA/prefix-length> <any|host <DA>|DA/prefix-length> [dscp <tos-bits|precedence>] [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`, **Resequencing the ACEs in an IPv6 ACL** on page 101.

NOTE: To insert a new ACE between two existing ACEs in an ACL, precede `deny` or `permit` with an appropriate sequence number. See **Inserting an ACE in an existing ACL** on page 99.

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.

`<ipv6|ipv6-protocol|ipv6-protocol-nbr>`

ipv6 - Any IPv6 packet.

ipv6-protocol -Any one of the following IPv6 protocol names:

esp
ah
sctp
icmp*
tcp*
udp*

*For TCP, UDP, and ICMP, additional, optional criteria can be specified, as described in **Options for TCP and UDP traffic in IPv6 ACLs** on page 95 and subsequent sections.

ipv6-protocol-nbr -The protocol number of an IPv6 packet type, such as “8” for Exterior Gateway Protocol or 121 for Simple Message Protocol. (Range: 0–255)

(For a listing of IPv6 protocol numbers and their corresponding protocol names, refer to the IANA protocol number assignments at **www.iana.com**.)

`<any|host <SA>|SA/<prefix-length>`

This is the first instance of IPv6 addressing in an ACE. It follows the protocol specifier and defines the source IPv6 address (SA) a packet must carry for a match with the ACE.

`any` -Allows IPv6 packets from any IPv6 SA.

`host <SA>` - Specifies only packets having a single address as the SA. Use this criterion when you want to match only the IPv6 packets from a single SA.

SA prefix-length - Specifies packets received from 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. (See **Using CIDR notation to enter the IPv6 ACL prefix length** on page 88.) In a

given ACE, the SA prefix length defines how many leftmost bits in a packet's SA must exactly match the SA configured in the ACE.

Prefix-length applications

- 2001:db8:0:e102::10:100/120 matches any IPv6 address in the range of 2001:db8:0:e102::10:<0100 - 01FF>
- 2001:db8:a0:e102::/64 matches any IPv6 address having a prefix of 2001:db8:a0:e102.
- FE80::/16 matches any link-local address on an interface.

NOTE: For more information on how prefix lengths are used in IPv6 ACLs, see **How an ACE uses a mask to screen packets for matches** on page 81.

`<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. (See **Using CIDR notation to enter the IPv6 ACL prefix length** on page 88.) 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.

`[dscp <codepoint|precedence]`

This option follows the DA to include a DSCP codepoint or precedence as a matching criteria.

`codepoint` : Supports these codepoint selection options:

`0-63`: Select a specific DSCP codepoint by entering its decimal equivalent.

Assured Forwarding (AF) codepoint matches:	
AF	**DSCPMatch**
af11	001010
af12	001100
af13	001110
af21	010010
af22	010100
af23	010110
af31	011010

Table Continued

Assured Forwarding (AF) codepoint matches:	
AF	**DSCPMatch**
af32	011100
af33	011110
af41	100010
af42	100100
af43	100110

`default`: Matches with the 000000 (default) DSCP.

`ef`: Expedited forwarding (EF; 000000) DSCP match.

`precedence`: Supports selection of a precedence setting in the DSCP.

Option	Precedence Bits	Name
cs1	001	priority
cs2	010	immediate
cs3	011	flash
cs4	100	flash-override
cs5	101	critical
cs6	110	internet (for internetwork control)
cs7	111	network (for network control)

NOTE: The precedence criteria described in this section are applied in addition to any other selection criteria configured in the same ACE. Also, where `dscp` is configured in a given ACE, the `established` keyword and the optional TCP control bits cannot be configured.

`[dscp <`*`codepoint`*`|`*`precedence`*`] [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. (See **Enabling ACL logging on the switch** on page 114.)

For a given ACE, if `log` is used, it must be the last keyword entered.

Table 10: *DSCP codepoints with decimal equivalents*

DSCP bits	Decimal
000000	0 (default)
000001	1
000010	2
000011	3
000100	4
000101	5
000110	6
000111	7
001000	8
001001	9
001010	10 (1)
001011	11
001100	12 (1^1)
001101	13
001110	14 (2^1)
001111	15
010000	16
010001	17
010010	18 (0^1)
010011	19
010100	20 (0^1)
010101	21
010110	22 (3^1)
010111	23

Table Continued

DSCP bits	Decimal
011000	24
011001	25
011010	26 (4[1])
011011	27
011100	28 (4[1])
011101	29
011110	30 (5[1])
011111	31
100000	32
100001	33
100010	34 (6[1])
100011	35
100100	36 (6[1])
100101	37
100110	38 (7[1])
100111	39
101000	40
101001	41
101010	42
101011	43
101100	44
101101	45
101110	46 (7)
101111	47
110000	48

Table Continued

DSCP bits	Decimal
110001	49
110010	50
110011	51
110100	52
110101	53
110110	54
110111	55
111000	56
111001	57
111010	58
111011	59
111100	60
111101	61
111110	62
111111	63

[1] Assured Forwarding codepoint and 802.1p precedence.

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, the destination, or both. Use of TCP criteria also allows the `established` option for controlling TCP connection traffic.

Configuring TCP

Syntax:

```
<deny|permit> tcp
<SA> [comparison-operator <tcp-src-port>]
<DA> [comparison-operator <tcp-dest-port>]
[established]
[ack][fin][rst][syn]
```

Configuring UDP

Syntax:

```
<deny|permit> udp
```

<SA> [*comparison-operator <udp-src-port>*]

<DA> [*comparison-operator <udp-dest-port>*]

Comparison operators for TCP or UDP

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.

Applying comparison operators

```
#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:

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, see **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.
2. Enter the port number or a well-known port name.

These are the same as those used with the TCP/UDP source-port options and are listed earlier in this command description.

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 prevents Telnet sessions in either direction, because responses to outbound requests are blocked. However, by using the established option, inbound Telnet traffic arriving in response to outbound Telnet requests are permitted, but inbound Telnet traffic trying to establish a new connection is 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 information on using TCP control bits, see RFC 793.

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.

NOTE:

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, see **Creating or editing ACLs offline** on page 111.

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-#>`
- 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. The following `show run` output shows an ACL named "My-list" using the default numbering scheme:

Default sequential numbering for ACEs

```
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
```

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:

Ways to append a new ACE to the end of an ACL

```
Switch(config)# ipv6 access-list My-list permit esp host 2001:db8:0:5ad::19 any [1]
Switch(Config)# ipv6 access-list My-list [2]
Switch(config-ipv6-acl)# permit ipv6 any host 2001:db8:0:5ad::1
```

[1] From the global configuration prompt, appends an ACE to the end of the ACL named “My-list”

[2] Enters the context of the “My-list” ACL and appends an ACE to the end of the list

Appending an ACE to an existing list

```
Switch(config-ipv6-acl)# deny ipv6 2001:db8:0:5ad::/64 any [1]
Switch (config-ipv6-acl)# permit ipv6 any any [2]
Switch(config-ipv6-acl)# show run
```

```
. . .
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
     40 permit icmp 2001:db8:0:5ad::/64 ::/0
     50 permit 50 2001:db8:0:5ad::19/128 ::/0
     60 permit ipv6 ::/0 2001:db8:0:5ad::1/128
     70 deny ipv6 2001:db8:0:5ad::/64 ::/0
     80 permit ipv6 ::/0 ::/0
   exit
```

[1] ACE appended as line 70

[2] Appended as line 80

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.

Inserting a new ACE in an existing ACL (examples)

From the global configuration context, insert a new ACE with a sequence number of 45 between the ACEs numbered 40 and 50 in **Appending an ACE to an existing list** on page 98.

Inserting an ACE in an existing ACL

```
Switch(Config)# ipv6 access-list My-list [1]
Switch(config-ipv6-acl)# 45 permit icmp host 2001:db8:0:5ad::33 ::/0 [2]
Switch(config-ipv6-acl)# show run
. . .
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
    40 permit icmp 2001:db8:0:5ad::/64 ::/0
    45 permit icmp 2001:db8:0:5ad::33 ::/0
    50 permit icmp 2001:db8:0:5ad::19/128 ::/0
    60 permit ipv6 ::/0 2001:db8:0:5ad::1/128
    70 deny ipv6 2001:db8:0:5ad::/64 ::/0
    80 permit ipv6 ::/0 ::/0
  exit
```

[1] Enters the Named-ACL context for “My-list”

[2] Inserts a new ACE assigned to line 45

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.

Inserting an ACE into an existing sequence

```
Switch(config)# Port_1_5400(config)# ipv6 access-list List-01 [1]

Switch(config-ipv6-acl)# permit ipv6 host fe80::100 host fe80::200 [2]
Switch(config-ipv6-acl)# permit ipv6 host fe80::103 any
Switch(config-ipv6-acl)# 11 permit ipv6 host fe80::110 host fe80:: [3]

Switch(config-ipv6-acl)# show run
Running configuration:
. . .
ipv6 access-list "List-01"
    10 permit ipv6 fe80::100/128 fe80::200/128
    11 permit ipv6 fe80::110/128 fe80::210/128
    20 permit ipv6 fe80::103/128 ::/0
  exit
```

[1] Becomes Line 10

[2] Becomes Line 20

[3] Lines 10 and 20 were automatically numbered according to their order of entry in the list. Line 11 was explicitly numbered by the `11 permit` command and was inserted in its proper place in the list.

Deleting an ACE from an existing 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.

The example below illustrates the process for deleting an ACE from a list:

Deleting an ACE from an IPv6 ACL

```
Switch(config)# show access-list My-List config

ipv6 access-list "My-List" [1]
    10 permit ipv6 fe80::100/128 ::/0
    20 deny ipv6 fe80::110/128 fe80::/124
    30 deny ipv6 fe80::111/128 fe80::/124
    40 permit ipv6 ::/0 ::/0
  exit
Switch(config)# ipv6 access-list My-List [2]
Switch(config-ipv6-acl)# no 30 [3]
Switch(config-ipv6-acl)# show access-list My-List config
```

```
ipv6 access-list "My-List" [4]
    10 permit ipv6 fe80::100/128 ::/0
    20 deny ipv6 fe80::110/128 fe80::/124
    40 permit ipv6 ::/0 ::/0 [5]
  exit
```

[1] ACL before deleting an ACE

[2] Enters the IPv6 ACL (**config-ipv6-acl**) context for “My-List”

[3] This command deletes the ACE at line 30

[4] ACL after deleting the ACE at Line 20

[5] The ACE at line 30 has been removed

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.

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.

Viewing and resequencing an ACL

```
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

Switch(config)# ipv6 access-list resequence My-List 100
100
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
```

This example resequences the “My-List” ACL at the bottom, so that the list begins with line 100 and uses a sequence interval of 100.

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.

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-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, 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, 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:

Procedure

1. Enter the remark first.
2. Then enter the related ACE. This results in the remark and the subsequent ACE having the same sequence number.

To append an ACE with an associated remark to the end of an ACL named "List-100," enter remarks from the **CLI** context for the desired ACL:

```
Switch(config)# ipv6 access-list List-100
Switch(config-ipv6-acl)# permit tcp host 2001:db8:0:b::100:17 eq telnet any
Switch(config-ipv6-acl)# permit tcp host 2001:db8:0:b::100:23 eq telnet any
Switch(config-ipv6-acl)# remark “BLOCKS UNAUTH TELNET TRAFFIC FROM SUBNET B”
Switch(config-ipv6-acll)# deny tcp 2001:db8:0:a::/64 eq telnet any
```

```
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
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.

Inserting remarks and related ACEs within an existing list

To insert an ACE with a remark within an ACL by specifying a sequence number:

Procedure

1. Insert the numbered remark first
2. Then, using the same sequence number, insert the ACE.

```
Switch(config-ipv6-acl)# 15 remark "PERMIT HTTP; STATION 23; SUBNET 1D"
Switch(config-ipv6-acl)# 15 permit tcp host 2001:db8:0:1d::23 eq 80 2001:db8:0:2f::/64


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

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

Procedure

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.
3. This step overwrites the former remark text with the new remark text.

To change the text of the remark at line 15 in **Inserting remarks and related ACEs within an existing list** on page 103 to "PERMIT HTTP FROM ONE STATION", use the following command:

```
Switch(config)# ipv6 access-list List-105
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:

Procedure

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, stand-alone 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
Switch# ipv6 access-list resequence XYZ 100 10
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.

Overwriting one remark with another

```
Switch(config-ipv6-acl)# permit ipv6 host fe80::a1:121 fe80::/104
Switch(config-ipv6-acl)# deny tcp any eq ftp 2001:db8:0:a1::/64
Switch(config-ipv6-acl)# remark Marketing
Switch(config-ipv6-acl)# remark Channel_Mktg
Port_1_5400(config-ipv6-acl)# show access-list Accounting config

ipv6 access-list "Accounting"
    10 permit ipv6 fe80::a1:121/128 fe80::/104
    20 deny tcp ::/0 eq 21 2001:db8:0:a1::/64
    30 remark "Channel_Mktg"
  exit
```

NOTE: Where multiple remarks are sequentially entered for automatic inclusion at the end of an ACL, each successive remark replaces the previous one until an ACE is configured for automatic inclusion at the end of the list.

Viewing ACL configuration data

This section describes how to view ACL summary and information related to VLAN ACLs.

Viewing an ACL summary

Lists the configured IPv4 and IPv6 ACLs, regardless of whether they are assigned to any interfaces.

Syntax:

```
show access-list
```

```
Switch(eth-Trk1)# sh access-list

Access Control Lists

  deny-fragmented-tcp-header            : Disabled
  deny-non-classifiable-layer4-header   : Disabled
  ACL Grouping                          : Disabled

  Type  Appl Name
  ----- ---- ------------------------
  ext   yes 101 [1]
  std   yes 55
  ext   yes Marketing
  ipv6  no  Accounting [2]
  ipv6  no  List-01-Inbound
  ipv6  yes List-02-Outbound
  ipv6  yes Test-1
```

[1] IPv4

[2] These ACLs exist in the configuration but are not applied to any interfaces and thus do not affect traffic

Summary table of access lists

Lists a summary table of the name, type, and application status of all ACLs (IPv4 and IPv6) configured on the switch.

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) • `ext` (Extended; protocol, 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.

Viewing the content of all ACLs on the switch

Lists the configuration details for every IPv4 and IPv6 ACL in the running-config file, regardless of whether any are actually assigned to filter traffic on specific interfaces.

Syntax:

```
show access-list config
```

Lists the configured syntax for all IPv4 and IPv6 ACLs currently configured on the switch.

NOTE: 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. See **Creating or editing ACLs offline** on page 111.

This information also appears in the `show running` output. If you execute `write memory` after configuring an ACL, it appears in the `show config` output.

An ACL configured syntax listing

```
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
```

The following example shows the ACLs on a switch configured with two IPv6 ACLs named “Accounting” and “List-01-Inbound”, and one extended IPv4 ACL named “101”:

An ACL configured syntax listing

```
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
ipv6 access-list "List-01-Inbound"
     10 permit icmp fe80::10:60/128 ::/0 dscp 38
     20 permit icmp fe80::10:77/128 ::/0 dscp 38
     30 permit icmp fe80::10:83/128 ::/0 dscp 38
     40 deny icmp ::/0 ::/0 dscp 38
     50 permit ipv6 fe80::10/112 ::/0
     60 deny ipv6 fe80::/64 ::/0
   exit
```

Viewing static port (and trunk) ACL assignments

Lists the identification and types 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 is in the `show running` output. Run the write memory command `write memory` after configuring an ACL in the `show config` output.

For example, the following output shows IPv4 and IPv6 ACLs configured on various ports and trunks on the switch:

Listing the ACL assignments for ports and trunks

```
Switch(config)# show access-list ports all

Access Lists for Port 1 [1]
  IPv6 Inbound    : test
  IPv6 Outbound   : test

Access Lists for Port Trk1 [2]
  IPv6 Inbound    : test
  IPv6 Outbound   : test
```

[1] An IPv6 ACL is filtering inbound and outbound traffic on port 1

[2] An IPv6 ACL is filtering inbound and outbound IPv6 traffic on Trunk 1(Trk1)

Viewing the content of a specific ACL

Displays a specific IPv6 or IPv4 ACL configured in the running config file in an easy-to-read tabular format.

Syntax:

`show access-list <identifier> [config]`

Displays detailed information on the content of a specific ACL configured in the running-config file.

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, see the latest version of the *ArubaOS-Switch Access Security Guide* for your switch.

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:

Listing an IPv6 ACL

```
Switch(config)# show access-list Accounting

Access Control Lists

  Name: Accounting
  Type: ipv6
  Applied: Yes

 SEQ  Entry
-------------------------------------------------------
 10   Action: permit
      Remark: Telnet Allowed [1]
      Src IP: 2001:db8:0:1af::10:14
      Prefix Len: 128 [2]
      Dst IP: :: [3]
      Prefix Len: 0
      Src Port(s): [4] Dst Port(s): eq 23 [5]
      Proto : TCP Option(s):[6]
      Dscp : -[7]

 20   Action: permit
      Src IP: 2001:db8:0:1af::10:23 [8]     Prefix Len: 128
      Dst IP: :: [9]                        Prefix Len: 0
      Src Port(s):  Dst Port(s): eq 23
      Proto : TCP Option(s):
      Dscp : -
```

```
30   Action: deny (log)
     Src IP: 2001:db8:0:1af::10        Prefix Len: 116
     Dst IP: ::                        Prefix Len: 0
     Src Port(s):  Dst Port(s):
     Proto : TCP Option(s):
     Dscp : -
```

[1] Indicates whether the ACL is applied to an interface

[2] Remark Field (appears if remark configured)

[3] Source Address

[4] Destination Address

[5] TCP Destination Port (Note: An empty TCP field indicates that the TCP port number for that field can be any value)

[6] Source and Destination Prefix Lengths

[7] TCP Source Port

[8] Protocol Data

[9] DSCP Codepoint or Precedence

Listing an IPv4 extended ACL

```
Switch(config)# show access-list List-120

Access Control Lists

  Name: List-120
  Type: Extended
  Applied: No [1]

 SEQ Entry
-------------------------------------------------------------
 10  Action: permit
     Remark: Telnet Allowed [2]
     Src IP: 10.30.133.27 [3]    Mask: 0.0.0.0          Port(s): eq 23
     Dst IP: 0.0.0.0 [4]         Mask: 255.255.255.255  Port(s):  [5]
     Proto : IP [6]
     TOS : -                     Precedence: - [7]

 20  Action: deny (log)
     Src IP: 10.30.133.1         Mask: 0.0.0.255        Port(s):
     Dst IP: 0.0.0.0             Mask: 255.255.255.255  Port(s):
     Proto : IP
     TOS : -                     Precedence: -

 30  Action: permit
     Src IP: 0.0.0.0             Mask: 255.255.255.255  Port(s):
     Dst IP: 0.0.0.0             Mask: 255.255.255.255  Port(s):
     Proto : IP
     TOS : -                     Precedence: -
```

[1] Indicates whether the ACL is applied to an interface

[2] Remark Field (Appears if remark configured)

[3] Source Address

[4] TCP Source Port

[5] Protocol Data

[6] Empty field indicates that the destination TCP port can be any value

[7] DSCP Codepoint and Precedence Data

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.

An ACL listed with the `config` option

```
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
```

Table 11: *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 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.

Table Continued

Field	Description
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.
Proto	Used in IPv6 ACEs and IPv4 extended ACEs to specify the packet protocol type to filter.
Port(s)	Used in IPv4 extended ACEs to show any TCP or UDP operator and port numbers 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 numbers included in the ACE.
DSCP	Used in IPv6 ACEs to show the DSCP precedence or codepoint setting, if any.
TOS	Used in IPv4 extended ACEs to indicate Type-of-Service setting, if any.
Precedence	Used in IPv4 extended ACEs to indicate the IP precedence setting, if any.

Viewing 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 **Editing an existing ACL** on page 98 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.

The offline process

Procedure

1. Begin by doing one of the following:

   a. 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:

   ```
   Switch# copy command-output 'show access-list config'
   tftp fe80::2a1:200 acl-001.txt pc
   ```

   b. 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 ACLs in the `*.txt` ASCII file format.

3. 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 [1]
   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
   ```

   [1] 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.

4. Use `copy tftp command-file` to download the file as a list of commands to the switch.
   Using `copy tftp command-file` to configure an ACL in the switch

   ```
   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.

5. 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, the next step is to manually assign the new ACL to the intended VLAN: `vlan <vid> ipv6 access-group <identifier> <vlan-in|vlan-out>`

6. 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. Verifying the `.txt` file download to the switch:

```
Switch(config)# show run
. . .
ipv6 access-list "acl-001"
   10 remark "Telnet Denied Here"
   10 deny tcp ::/0 ::/0 eq 23
   30 deny tcp ::/0 ::/0 log
   40 deny icmp ::/0 ::/0 134
   50 deny icmp ::/0 ::/0 133
   60 permit ipv6 ::/0 ::/0
   exit
. . .
vlan 20 [1]
   ipv6 access-group "acl-001" vlan-in
   exit
. . .
```

[1] As a part of the instruction set included in the .txt file, the ACL is assigned to inbound IP traffic on VLAN 20

NOTE: The comment preceded by " ; " in the .txt source file for this configuration do not appear in the ACL configured in the switch

7. If the configuration appears satisfactory, save it to the startup-config file:

```
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 both "permit" and "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 or outbound 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.
- If the RACL application is used, then IPv6 routing must be enabled on the switch.
- 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 114.

ACL logging operation

When the switch detects a packet match with an ACE and the ACE includes 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 the following example:

Content of messages generated by an ACL-deny action

Example of subsequent `deny` events detected by the switch for the same ACE:

```
ACL 12/01/08 10:04:45 List NO-TELNET, seq#10 denied tcp 2001:db8:0:1ae::1a:3(1612)
   2001:db8:0:1ad::1a:2(23) on vlan 1, port A7
```

Example Syslog report of the first `deny` event detected by the switch for this ACE.

```
Dec 1 10:04:45 2008:db8:0:1ad::1a:1 ACL:
ACL 12/01/08 10:04:45 : ACL NO-TELNET seq#10 denied 6 packets
```

Enabling ACL logging on the switch

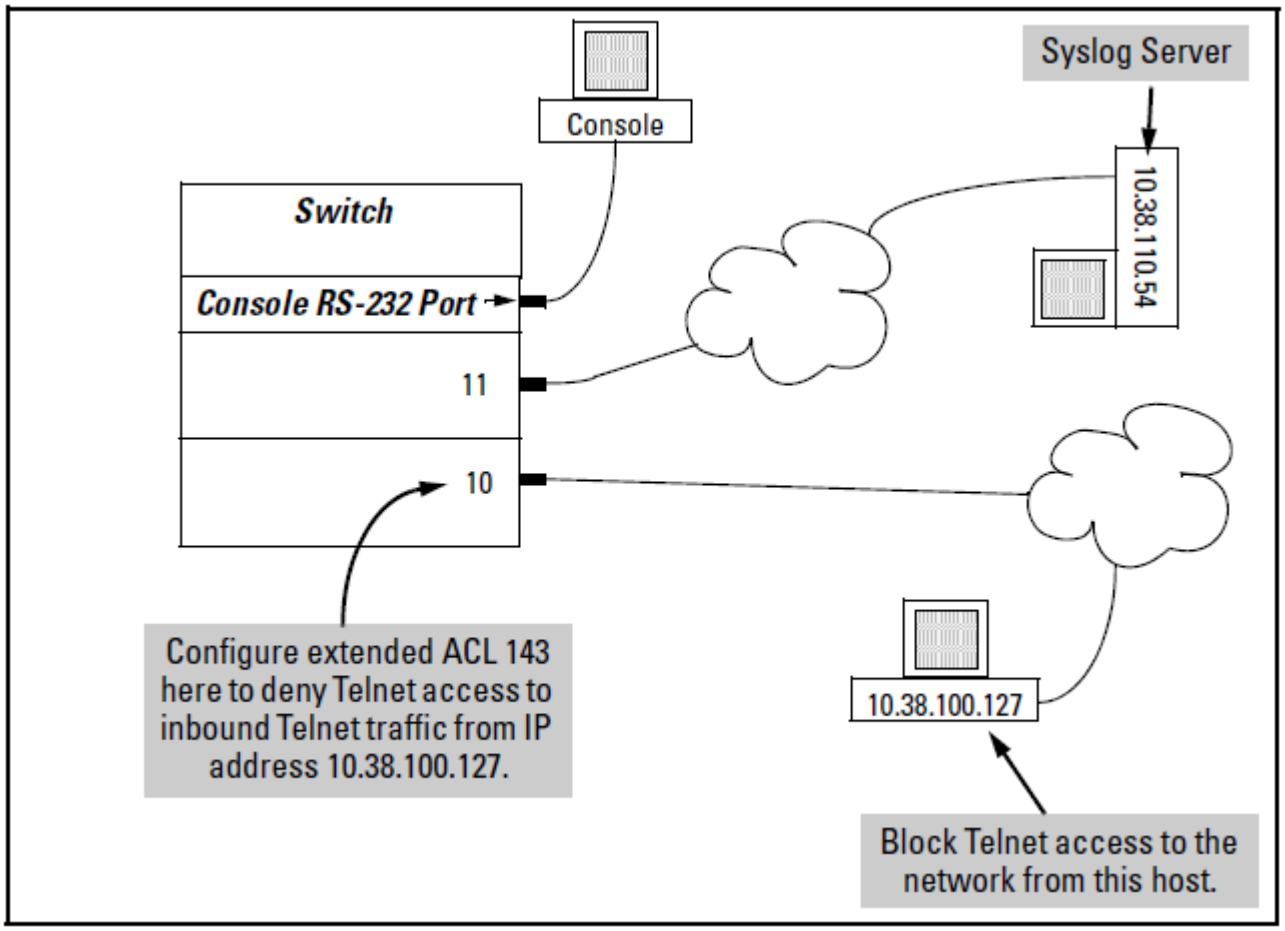
Procedure

1. If you are using a syslog server, use the `logging <ip-addr>` command to configure the syslog server IP addresses; ensure that the switch can access any syslog servers 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.
4. Destination options include `logging` and `session`. For more information on debug, see "debug and syslog messaging operation" in the latest management and configuration guide for your switch.
5. Use `debug acl` or `debug all` to configure the debug operation to include ACL messages.
6. 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 7: *Example of an ACL log application*



Commands for applying an ACL with logging:

```
Switch(config)# access-list 143 deny tcp host 10.38.100.127 any eq telnet
log
Switch(config)# access-list 143 permit ip any any
Switch(config)# interface 10 access-group 143 in
Switch(config)# logging 10.38.110.54
Switch(config)# debug ac1
Switch(config)# debug destination logging
Switch(config)# debug destination session
Switch(config)# write memory

Switch(config)# show debug
Debug Logging
 Destination:
  Logging
   10.38.110.54
  Session
 Enabled debug types:
```

```
event
acl log
```

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.
- ACL screening of IPv6 traffic generated by the switch. Outbound IPv6 RACL applications on a switch do not screen IPv6 traffic (such as broadcasts, Telnet, Ping, and ICMP replies) generated by the switch itself. Note that all ACLs applied on the switch do screen this type of traffic when other devices generate it. Similarly, all ACL applications can screen responses from other devices to unscreened IPv6 traffic the switch generates.
- ACL 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, Hewlett Packard Enterprise 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.)
  - When configuring logging, you can reduce excessive resource use by configuring the appropriate ACEs to match with specific hosts instead of entire subnets.
- 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, see "Monitoring Resources" in the latest management and configuration guide for your switch. See also the section "scalability and system maximums" in the same guide.
- Protocol support. ACL criteria does not include use of MAC address information or QoS.
- 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. 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:

Procedure

1. In the **interface** context, use the `no ipv6 access-group` command to remove the ACL from the interface.
2. Use the `no ipv6 access-list <name-str>` command to delete the ACL.

This section describes IPv6 router advertisements (RAs).

Overview of IPv6 RA

The routing switches covered by this guide support IPv6 RA configuration and transmission based on RFC 4861, “Neighbor Discovery for IP Version 6 (IPv6)” and RFC 4862, “IPv6 Stateless Address Autoconfiguration”.

IPv6 RAs on a VLAN provide the neighbor discovery policy the system administrator has configured for devices running in IPv6 host mode with address autoconfiguration enabled. RAs also enable hosts on a VLAN to build a list of default (reachable) routers on that VLAN.

RA general operation

An IPv6 routing switch configured as a member of a given VLAN transmits RAs for use by hosts on the VLAN. It also transmits unscheduled RAs in response to router solicitations received from IPv6 hosts on the VLAN. The values a host receives in an RA are applied to settings that have not already been configured on the host by the system operator. (Values in an RA can also replace host settings that were learned from a previous RA.)

When IPv6 unicast routing is enabled, RAs are transmitted by default on VLANs enabled for IPv6 and configured with an IPv6 link-local address, unless RA transmission has been explicitly suppressed. RA configuration includes:

Advertisement Value	Default	Page
managed flag (M-bit)	Not set	**VLAN context Neighbor Discovery (ND) configuration** on page 122
other-config-flag (O-bit)	Not set	**VLAN context Neighbor Discovery (ND) configuration** on page 122
prefix	The prefix of any global unicast IPv6 address configured on the VLAN interface[1]	**Configuring the global unicast prefix and lifetime for hosts on a VLAN** on page 125
length	N/A; based on existing configuration	—
valid lifetime	2,592,000 seconds (30 days)	—
preferred lifetime	604,800 seconds (seven days)	—
autoconfig (A-bit)	Set (host autoconfig enabled)	—
on-link (L-bit)	Set (use prefix on subject VLAN)	—
RA transmission interval		—

Table Continued

Advertisement Value		Default	Page
	maximum	600 seconds	**Configuring the range for intervals between RA transmissions on a VLAN** on page 123
	minimum	200 seconds	**Configuring the range for intervals between RA transmissions on a VLAN** on page 123
current hop limit		64	**Setting or changing the hop-limit for host-generated packets** on page 123
default lifetime		1800 seconds (3 x max. transmission interval)	**Setting or changing the default router lifetime** on page 123
reachable time		Unspecified (0)	**Changing the reachable time duration for neighbors** on page 124
retransmission timer		Unspecified (0)	**Setting or changing the neighbor discovery retransmit timer** on page 124

[1] Default operation excludes prefixes of stateless autoconfigured addresses.

RA basics

- Enabling IPv6 unicast routing on a routing switch initiates transmission of RAs on active, IPv6-enabled VLANs unless RA transmission has been suppressed.
- RAs are not routed.
- A host response to an RA depends on how the host implements IPv6. Generally, settings in an RA received by a host replaces settings received from an earlier RA. Settings configured directly on a host by an operator may override values received in an RA for the same settings.
- When a host receives a default "unspecified" value in an RA, the host applies either its own current setting for that value, or the defaults specified in RFC 4861 or other applicable RFCs, depending on how IPv6 is implemented in the host.
- The M-bit and O-bit flags enable RAs to be configured either to act as the sole source of host addressing and related settings, or to direct the host to use a DHCPv6 server for some or all such settings.

Setting up your IPv6 RA policy

- Is there a role for a DHCPv6 server in host configuration on a given VLAN, and what host services and policy will be configured? Affects M-bit and O-bit options (page **VLAN context Neighbor Discovery (ND) configuration** on page 122)
- What is the ND policy that should be advertised? Includes hop-limit for host-generated traffic, the default router period, neighbor reachable time, and retransmit time for neighbor solicitations.

- What prefixes should be advertised, and what prefixes should be suppressed? Prefixes configured on the routing switch VLAN interface will be included in RAs on that VLAN unless specifically denied.
- What should be the maximum and minimum intervals (in seconds) for transmitting RAs?
- Are there any VLANs on the routing switch where RAs should be suppressed?
- Will multiple routing devices be used to send RAs on a VLAN?
  - The first RA received by a host determines the default router for that host. Other routers included in subsequent RAs received by the host become backup default routers for that host.
  - What, if any, differences are acceptable in RAs from different routing devices?

Configuring IPv6 RAs

When IPv6 unicast routing is enabled on the routing switch, RAs are transmitted on all IPv6-enabled VLANs unless explicitly suppressed globally or per-VLAN.

The following steps provide a general outline of the steps for configuring the routing switch for nondefault RA operation on all IPv6-enabled VLANs:

Procedure

1. Enable IPv6 routing on your network.
2. Enable IPv6 unicast routing. (This must be enabled to allow configuration of other routing protocols).

   ```
   Switch(config)# ipv6 unicast-routing
   ```

   (This command enables RA transmission on any VLAN where RAs are not specifically suppressed.)
3. Configure the desired per-VLAN RA operation:

   a. Use the M-bit and O-bit settings to specify the source for IPv6 host configuration; see page **VLAN context Neighbor Discovery (ND) configuration** on page 122:

      I. M-bit setting:

         - Get configuration from RAs (default).
         - Get configuration from DHCPv6.

      II. O-bit setting (applies only if M-bit setting is left in default state):

         - Use RA source for global unicast prefixes (default).
         - Do not use the RA for nonprefix configuration.

   b. Configure global unicast prefix assignments; see **Configuring the global unicast prefix and lifetime for hosts on a VLAN** on page 125:

      I. Specify any prefixes not configured on the routing switch VLAN interface that should be transmitted in RAs to IPv6 hosts on the VLAN.

      II. Deny any prefixes configured on the routing switch VLAN interface that should not be transmitted in RAs to IPv6 hosts on the VLAN. (Default: Global unicast prefixes configured on the routing switch VLAN interface are included in RAs.)

c. Configure the maximum and minimum interval for transmitting RAs on the VLAN; see page **Configuring the range for intervals between RA transmissions on a VLAN** on page 123.

NOTE:

The routing switch also transmits RAs when it receives router solicitations from a host. Autoconfiguration must be enabled on the host before it will generate router solicitations on the VLAN.

d. Configure the ND policy for hosts on the VLAN to use:

   I. hop-limit (default: 64; see page **Setting or changing the hop-limit for host-generated packets** on page 123)

   II. Default router lifetime (default: 1800 seconds; see page **Setting or changing the default router lifetime** on page 123)

   III. Reachable time duration to advertise for confirmed neighbors (default: unspecified (0); see page **Changing the reachable time duration for neighbors** on page 124)

   IV. Retransmit time to advertise for neighbor solicitations (default: unspecified (0); see page **Setting or changing the neighbor discovery retransmit timer** on page 124)

e. Configure per-VLAN RA suppression for any VLAN on which you do not want the routing switch to transmit RAs. (See **Viewing the RA configuration** on page 130.) `Switch(vlan-1)# ipv6 nd ra suppress`

Configuring RAs on multiple switches with a common VLAN

Multiple routing switches transmitting RAs on the same VLAN can provide redundancy. Typically, a host identifies the first router from which it receives an RA as the default router. The host uses any RAs received later from other routers to identify backup default routers.

While advertised prefixes can be different, the per-VLAN RA policy should be the same for all routers transmitting RAs on a given VLAN. This includes the following parameters:

- managed-config-flag (M-bit)
- other-config-flag (O-bit)
- default router lifetime
- hop-limit
- reachable-time for neighbors
- retransmit time for neighbor solicitations

Global configuration context commands

This section describes commands to enable or disable IPv6 RA generation and IPv6 routing.

Enabling or disabling IPv6 RA generation

Syntax:

```
ipv6 nd suppress-ra
```

```
no ipv6 nd suppress-ra
```

Global config command to suppress transmission of IPv6 RAs on all VLANs configured on the routing switch. Overrides RAs enabled per-VLAN.

The `no` form of the command globally disables RA suppression. Note that globally enabling RAs on the routing switch does not override per-VLAN RA suppression (using the `ipv6 nd ra suppress` command in a **VLAN** context). See **Suppressing RAs on a VLAN** on page 128.

Default: RA suppression disabled

Enabling or disabling IPv6 routing

Syntax:

```
ipv6 unicast-routing
```

```
no ipv6 unicast-routing
```

Global config command to enable or disable IPv6 routing. Must be enabled for routing operation. Enabling IPv6 routing activates RA generation on VLANs unless RAs are suppressed globally or per-VLAN.

The `no` form of the command disables IPv6 routing and RAs on the routing switch.

Default: Disabled

VLAN context Neighbor Discovery (ND) configuration

This section describes commands for ND configuration.

Configuring DHCPv6 service requirements

Syntax:

```
ipv6 nd ra managed-config-flag
```

```
no ipv6 nd ra managed-config-flag
```

```
ipv6 nd ra other-config-flag
```

```
no ipv6 nd ra other-config-flag
```

`managed-config-flag` : Controls the M-bit setting in RAs the router transmits on the current VLAN. Enabling the M-bit directs clients to acquire their IPv6 addressing and ND host configuration information for the current VLAN interface from a DHCPv6 server.

- When the M-bit is enabled, receiving hosts ignore the `other-config-flag` (O-bit) setting described below.
- When the M-bit is disabled (the default), receiving hosts expect to receive their IPv6 addressing and ND configuration settings from the RA unless the O-bit is enabled.

`other-config-flag` : Ignored unless the M-bit (above) is disabled in RAs. Controls the O-bit in RAs the router transmits on the current VLAN.

Enabling the O-bit while the M-bit is disabled directs hosts on the VLAN to acquire their ND configuration settings from a DHCPv6 server and their global unicast prefixes from the RA.

The `no` form of either command turns off (disables) the setting for that command in RAs.

NOTE: In the default configuration, both the M-bit and the O-bit are disabled, and a host receiving the RA must acquire its prefix and ND configuration from the RA itself and not from a DHCPv6 server.

Default for both settings: Disabled

Configuring the range for intervals between RA transmissions on a VLAN

The interval between RA transmissions on a VLAN is a random value that changes every time an RA is sent. The interval is calculated to be a value between the current `max-interval` and `min-interval` settings described below.

Syntax:

```
ipv6 nd ra max-interval <4-1800>
no ipv6 nd ra max-interval <4-1800>
ipv6 nd ra min-interval <3-1350>
no ipv6 nd ra min-interval <3-1350>
```

VLAN context commands for changing the maximum and minimum intervals between transmissions of IPv6 RAs on the VLAN. These values have one setting per VLAN and do not apply to RAs sent in response to a router solicitation received from another device.

`max-interval` : Must be equal to or less than the configured lifetime setting. Attempting to set `max-interval` to a value greater than the configured lifetime setting results in an error message.

The `no` form of the `max-interval` command returns the setting to its default, provided the default value is less than or equal to 75% of the new maximum interval you are setting.

Attempting to set `max-interval` to a value that is not sufficiently larger than the current `min-interval` also results in an error message.

Default: 600 seconds; Range: 4–1800 seconds

`min-interval` : Must be less than or equal to 75% of `max-interval`. Attempting to set `min-interval` to a higher value results in an error message.

The `no` form of the `min-interval` command returns the setting to its default, provided the default value is less than or equal to 75% of the current `max-interval` setting.

Default: 200 seconds; Range: 3–1350 seconds

Setting or changing the hop-limit for host-generated packets

Syntax:

```
ipv6 nd ra hop-limit <0-255>
no ipv6 nd ra hop-limit <0-255>
```

`hop-limit` : **VLAN**-context command to specify the hop-limit a host includes in the packets it transmits.

A setting of **0** means the hop-limit is unspecified in the RAs originating on the current VLAN. In this case, the hop-limit is determined by the host.

The `no` form of the command resets the hop-limit to zero (unspecified), which eliminates the hop-limit from the RAs originating on the VLAN.

Default: 64; Range: unspecified 0 – 255

Setting or changing the default router lifetime

Syntax:

```
ipv6 nd ra lifetime <0-9000>
no ipv6 nd ra lifetime <0-9000>
```

`lifetime` : **VLAN**-context command for configuring the lifetime in seconds for the routing switch to be used as a default router by hosts on the current VLAN. This setting must be configured to a value greater than or equal to the `max-interval` setting.

A given host on a VLAN refreshes the default router lifetime for a specific router each time the host receives an RA from that router. A specific router ceases to be a default router candidate for a given host if the default router lifetime expires before the host is updated with a new RA from the router.

A setting of **0** (unspecified) for default router lifetime in an RA indicates that the routing switch is not a default router on the subject VLAN.

Default: Three times the `ra max-interval` setting. Range: unspecified 0 – 9000 seconds

Changing the reachable time duration for neighbors

Syntax:

```
ipv6 nd ra reachable-time <1000-3600000|unspecified>
```

```
no ipv6 nd ra reachable-time <1000-3600000|unspecified>
```

`reachable-time` : **VLAN**-context command for all hosts on the VLAN to configure as the reachable time duration for a given neighbor after receiving a reachability confirmation from the neighbor. This value is used to ensure a uniform reachable time among hosts on the VLAN by replacing the individually configured settings on various hosts on the VLAN.

`1000-3600000` : Reachable time in milliseconds.

`unspecified` : Configures the reachable time to zero, which disables the reachable-time setting in RAs on the current VLAN.

The `no` form also disables the reachable-time setting in RAs on the current VLAN.

Default: unspecified (0); Range: 1000–3600000 ms

NOTE: If multiple routers on the same VLAN are configured to advertise a reachable time, all such routers should use the same reachable-time setting.

Setting or changing the neighbor discovery retransmit timer

Syntax

```
ipv6 nd ra NS-interval <1000-4294967295|unspecified>
```

```
ipv6 nd ra NS-interval
```

```
no ipv6 nd ra NS-interval
```

Used on VLAN interfaces to advertise the period (retransmit timer) in milliseconds between ND solicitations sent by a host for an unresolved destination, or between DAD neighbor solicitation requests. Increasing this setting is indicated where neighbor solicitation retries or failures are occurring, or in a "slow" (WAN) network.

The `no` form returns the setting to its default.

Parameters

`1000-4294967295`

An advertised setting in this range replaces the corresponding, locally configured setting in hosts on the VLAN.

`unspecified`

Sets the retransmit timer value in RAs to zero, which causes the hosts on the VLAN to use their own locally configured NS-interval settings instead of using the value received in the RAs.

Default: unspecified (0) ; Range: 1000–4294967295 ms

NOTE: This is the retransmit timer advertised as a host-specific variable. It is separate from the retransmit timer used by the routing switch for its own ND solicitations (`ipv6 nd NS-interval`).

If multiple routers on the same VLAN are configured to advertise an `NS-interval` (retransmit time), all such routers should use the same NS-interval setting.

The default NS-interval setting for IPv6 host operation on devices is 1000 ms. When the above command is used with the `unspecified` option to configure RAs, host devices configured by using the RA maintain their preconfigured NS-interval settings.

Configuring the global unicast prefix and lifetime for hosts on a VLAN

These commands define the content of RAs transmitted on a VLAN.

Syntax:

`ipv6 nd ra prefix <ipv6-prefix|prefix-len> <<valid-lifetime> <preferred-lifetime> | at <valid-date> <preferred-date> infinite | no-advertise> [no-autoconfig] [off-link]`

`no ipv6 nd ra prefix <ipv6-prefix|prefix-len> <<valid-lifetime> <preferred-lifetime> | at <valid-date> <preferred-date> infinite | no-advertise> [no-autoconfig] [off-link]`

`ipv6 nd ra prefix default <<valid-lifetime> <preferred-lifetime> | at <valid-date> <preferred-date> | infinite | no-advertise> [no-autoconfig] [off-link]`

`no ipv6 nd ra prefix default <<valid-lifetime> <preferred-lifetime> | at <valid-date> <preferred-date> | infinite | no-advertise> [no-autoconfig] [off-link]`

Options for *<valid-lifetime> <preferred-lifetime>* :

Time in seconds:

`[<0-4294967295> <0-4294967295>]`

Specific date and time

`[ at <valid-lifetime> <preferred-lifetime> ]`

`<valid-lifetime-MM/DD[/[YY]YY]]>`

`<valid-lifetime-HH:MM[:SS]>`

`<preferred-lifetime-MM/DD[/[YY]YY]]>`

`<preferred-lifetime-HH:MM[:SS]>]`

`<valid-date> <preferred-date>`

VLAN-context command for specifying prefixes for the routing switch to include in RAs transmitted on the VLAN. IPv6 hosts use the prefixes in RAs to autoconfigure themselves with global unicast addresses. A host's autoconfigured address is composed of the advertised prefix and the interface identifier in the host's current link-local address.

valid-lifetime : The total time the prefix remains available before becoming unusable. After preferred-lifetime expiration, any autoconfigured address is deprecated and used only for transactions that began before the preferred-lifetime expired. If the valid lifetime also expires, the address becomes unusable. Default: 2,592,000 seconds–30 days; Range: 0–4294967295 seconds.

preferred-lifetime : The span of time during which the address can be freely used as a source and destination for traffic. This setting must be less than or equal to the corresponding `valid-lifetime` setting. Default: 604,000 seconds–7 days; Range: 0–4294967295 seconds

NOTE: The valid and preferred lifetimes designated in this command are fixed values. Each successive transmission of the same RA contains the same valid and preferred lifetimes.

For more information on valid and preferred lifetimes, see **Address lifetimes** on page 26.

`default` : Applied to all on-link prefixes that are not individually set by the`ipv6 ra prefix <ipv6-prefix|prefix-len>` command. It applies the same valid and preferred lifetimes, link state, autoconfiguration state, and advertise options to the advertisements sent for all on-link prefixes that are not individually configured with a unique lifetime. This also applies to the prefixes for any global unicast addresses configured later on the same VLAN.

Using `default` once, and then using it again with any new values results in the new values replacing the former values in advertisements.

If `default` is used without the `no-advertise`, `no-autoconfig`, or the `off-link` keyword, the advertisement setting for the absent keyword is returned to its default setting.

NOTE: To configure a prefix as `off-link` or `no-autoconfig`, you must enter unique valid and preferred lifetimes with the `prefix` command (instead of the `default` command).

`ipv6-prefix / prefix-len` : Specifies the prefixes to advertise on the subject VLAN. A separate instance of the command must be used for each prefix to advertise.

`infinite` : Specifies that the prefix lifetime will not expire. This option sets the valid and preferred lifetimes to infinity. (All bits set to 1; ffffffff.)

`no-advertise` : Specifies no advertisement for the prefix. For example, if the routing-switches VLAN interface is configured with any prefixes that you do not want advertised on the VLAN, use this command to specify the prefixes to withhold from advertisements on the subject VLAN. Default: Advertising enabled.

`no-autoconfig` : Disables host autoconfiguration by turning off the A-bit in RAs. This requires hosts to acquire prefixes through manual or DHCPv6 assignments. Depending on the host implementation, a host that was previously configured by an RA to use autoconfiguration will not be affected by a later RA that includes `no-autoconfig` (unless the host disconnects and reconnects to the network). To re-enable host autoconfiguration (turn on the A-bit in RAs) for a given RA, use `ipv6 nd ra prefix` again, without invoking `no-autoconfig`. Default: A-bit turned on— host autoconfig turned on.

`off-link` : Sets the (L-bit) prefix information in an RA to indicate that the advertised prefix is not on the subject VLAN. A host that was previously configured using an RA without `off-link` will not be affected by a later RA that includes `off-link` (unless the host disconnects and reconnects to the network). Can be used in instances where the prefix is being deprecated, and you do not want any newly brought up hosts to use the prefix. Default: L-bit turned off.

The `no` form of the command deletes the specified prefix from RAs.

Using the default command to configure prefix advertisement content (example)

The table below lists the global unicast addresses configured on a VLAN, with original and updated settings configured using the `default` command.

Address or prefix	Interface	Original lifetime & autoconfig	Updated lifetime & autoconfig	Advertise on VLAN 100?
2001:db8:0:f::f1/64	VLAN 100	15 days	30 days	Yes
2001:db8:0:b::b1/64	VLAN 100	14 days Auto: Yes	25 days Auto: No(	

Table Continued

Address or prefix	Interface	Original lifetime & autoconfig	Updated lifetime & autoconfig	Advertise on VLAN 100?
2001:db8:0:c::c1/64	VLAN 100	Set in the following example.	Changed in the following example.	
2001:db8:0:d::d1/64	VLAN 100			
2001:db8:0:a::/64	Off-Link	12/31/2010 at 00:00:01 12/20/2010 at 00:00:01 Auto: Yes	not updated	

Using the default command to configure and update prefix advertisements

```
Switch(config)# vlan 100
Switch(vlan-100)# ipv6 address 2001:db8:0:f::f1/64
Switch(vlan-100)# ipv6 address 2001:db8:0:b::b1/64[1]
Switch(vlan-100)# ipv6 address 2001:db8:0:c::c1/64
Switch(vlan-100)# ipv6 nd ra prefix default 1296000 1209600[2]
Switch(vlan-100)# show ipv6 nd ra prefix vlan 100

 IPv6 Neighbor Discovery Prefix Information

 VLAN Name : VLAN100[3]

   IPv6 Prefix        : Default
   Valid Lifetime     : 15 days
   Preferred Lifetime : 14 days
   On-link Flag       : On
   Autonomous Flag    : On
   Advertise Flag     : On

Switch(vlan-100)# ipv6 address 2001:db8:0:d::d1/64[4]
Switch(vlan-100)# ipv6 nd ra prefix 2001:db8:0:d::/64 infinite no-autoconfig
Switch(vlan-100)# ipv6 nd ra prefix 2001:db8:0:a::/64 at 12/31/2010 00:00:01 12/20/2010 00:00:01 off-link[5]
Switch(vlan-100)# show ipv6 nd ra prefix vlan 100

 IPv6 Neighbor Discovery Prefix Information

 VLAN Name : VLAN100

   IPv6 Prefix        : Default[6]
   Valid Lifetime     : 15 days
   Preferred Lifetime : 14 days
   On-link Flag       : On
   Autonomous Flag    : On
   Advertise Flag     : On

   IPv6 Prefix        : 2001:db8:0:a::/64[7]
   Valid Lifetime     : 12/31/2010 00:00:01
   Preferred Lifetime : 12/20/2010 00:00:01
   On-link Flag       : Off
   Autonomous Flag    : On
   Advertise Flag     : On

   IPv6 Prefix        : 2001:db8:0:d::/64[8]
   Valid Lifetime     : Infinite
   Preferred Lifetime : Infinite
   On-link Flag       : On
   Autonomous Flag    : Off
   Advertise Flag     : On

Switch(vlan-100)# ipv6 nd ra prefix default 2592000 2160000 no-autoconfig[9]
Switch(vlan-100)# show ipv6 nd ra prefix vlan 100

 IPv6 Neighbor Discovery Prefix Information

 VLAN Name : VLAN100
```

```
  IPv6 Prefix        : Default[10]
  Valid Lifetime     : 30 days
  Preferred Lifetime : 25 days
  On-link Flag       : On
  Autonomous Flag    : Off
  Advertise Flag     : On

  IPv6 Prefix        : 2001:db8:0:a::/64[11]
  Valid Lifetime     : 12/31/2010 00:00:01
  Preferred Lifetime : 12/20/2010 00:00:01
  On-link Flag       : Off
  Autonomous Flag    : On
  Advertise Flag     : On

  IPv6 Prefix        : 2001:db8:0:d::/64[12]
  Valid Lifetime     : Infinite
  Preferred Lifetime : Infinite
  On-link Flag       : On
  Autonomous Flag    : Off
  Advertise Flag     : On
```

[1] Global unicast addresses configured on VLAN 100

[2] To enable advertising prefixes of global unicast addresses configured on the VLAN, the `default` command sets default lifetime, prefix link status (on or off-link), autoconfiguration (Autonomous Flag) status (on or off), and advertisement setting (on or off).

NOTE: Applies only to prefixes in global unicast addresses configured on the VLAN and not uniquely configured by the `prefix` command.

[3] To enable advertising prefixes of global unicast addresses configured on the VLAN, the `default` command sets default lifetime, prefix link status (on or off-link), autoconfiguration (Autonomous Flag) status (on or off), and advertisement setting (on or off).

[4] Show command displays default prefix mode settings for global unicast addresses configured on VLAN 100

[5] New global unicast address configured on the VLAN. Followed by command to assign unique lifetime and autoconfig setting in the advertisements for this prefix. Link flag and Advertise flag omitted from the command and therefore set to “On” by default.

[6] Off-link prefix designated with unique lifetime. Autoconfig (Autonomous) flag and Advertise flag omitted from the command and therefore set to “On” default

[7] Show command displays default advertisement settings for prefixes of global unicast addresses configured on VLAN 100

[8] Show command displays unique advertisement settings for 2001:db8:0:a::/64 also configured on VLAN 100

[9] Show command displays unique advertisement settings for 2001:db8:0:d::/64 identified as an off-link prefix

[10] For prefixes configured on the VLAN and not specifically addressed by a prefix command, default changes the default lifetime and the autoconfig setting in advertisements for these prefixes. On-Link flag and Advertise flag omitted from the command and therefore set to “On” by default

[11] Show command displays changes in default prefix mode settings for global unicast addresses configured on VLAN 100

[12] No change for the on-link prefix specifically configured by a prefix command, and the off-link prefix that is also configured for advertisement on the VLAN

Suppressing RAs on a VLAN

Syntax:

```
ipv6 nd ra suppress

no ipv6 nd ra suppress
```

VLAN-context command to turn off (disable) transmission of RAs from the routing switch on the VLAN.

The `no` form of the command turns on (enables) RA transmission from the routing switch on the current VLAN.

Default: Suppression disable, that is, RA enabled on the VLAN.

Restricting IPv6 RAs

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

Only physical ports and trunk ports are 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:

`ipv6 ra-guard ports <port-list> [log]`

`no ipv6 ra-guard ports <port-list> [log]`

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

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

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

Enabling RA Guard

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

Operating notes for RA guard

- 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.
- 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.

Use the `show ipv6 ra-guard` command to display configuration and statistical information about RA Guard.

Configuration and statistics for RA Guard

```
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
```

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

Running config file showing line for RA-Guard

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

Running configuration:

; Jxxxxx Configuration Editor; Created on release #xx.16.xx.0000
; Ver #02.01.0f:0c

hostname "Switch"
module 1 type Jxxxxx
module 2 type Jxxxxx
module 3 type Jxxxxx
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 [1]
```

[1] RA Guard is enabled on port 6; logging is enabled.

Viewing the RA configuration

Syntax:

`show ipv6 nd ra`

`show ipv6 nd ra [prefix [vlan <vid>]]`

Without the optional keywords, this command displays the global and per-VLAN RA neighbor discovery configuration on a specific routing switch. This indicates the per-VLAN content of RAs transmitted from the routing switch.

`prefix` : Displays the prefixes, valid lifetime, and onlink/auto values advertised by the routing switch on all VLANs configured for RA operation.

`prefix vlan <vid>` : Displays values for each prefix configured using `ipv6 nd ra prefix` on the specified VLAN; see **Configuring the global unicast prefix and lifetime for hosts on a VLAN** on page 125.

`IPv6 Prefix` : Displays values for specific prefixes configured for RAs on a VLAN by the `ipv6 nd ra prefix` command, plus `Default` (to apply to any global unicast prefixes on the same VLAN(s) that have not been specifically configured by `ipv6 nd ra prefix`).

`Valid Lifetime` : The valid lifetime configured for the indicated prefix.

`Preferred Lifetime` : The preferred lifetime configured for the indicated prefix.

`On-link Flag` : Indicates whether the prefix is advertised as on-link. Default: On; On-link enabled.

`Autonomous Flag` : Indicates whether address autoconfiguration is turned on. Default: On; Autoconfiguration enabled.

`Advertise Flag` : Indicates whether advertisement for the subject prefix is turned on. Default: On.

General output listing the per-VLAN RA configuration on a routing switch

```
Switch(config)# show ipv6 nd ra

IPv6 Router Advertisement Configuration

Global RA Suppress   : No
Global Hop Limit     : 10
IPv6 Unicast Routing : Enabled

 VLAN Suppress Interval Lifetime Mngd Other RCH Time NS Intrvl Hop
  ID    RA     Min/Max   (sec)   Flag Flag    (ms)     (ms)   Limit
----- -------- -------- -------- ---- ----- -------- -------- -----
1     Yes      200/600  1800     No   No    0        0        10
22    No       200/600  1800     No   No    0        0        10
30    Yes      200/600  1000     No   No    0        0        4
```

Output for VLANs where specific prefixes have been configured for RAs

```
Switch(config)# show ipv6 nd ra prefix

  IPv6 Neighbor Discovery Prefix Information

VLAN Name : VLAN30

  IPv6 Prefix                      Valid Lifetime      Onlink/Auto
  -------------------------------- ------------------- -----------
  Default                          Infinite            On/On
  2001:db8:f:1b::/64               12/31/2010 00:00:01 Off/On
  2001:db8:f:1d::/64               12/31/2010 00:00:01 On/On
```

Detailed prefix configuration data for a specific VLAN

```
Switch(config)# show ipv6 nd ra prefix vlan 30

 IPv6 Neighbor Discovery Prefix Information

 VLAN Name : VLAN30

   IPv6 Prefix        : Default
   Valid Lifetime     : Infinite
   Preferred Lifetime : Infinite
   On-link Flag       : On
   Autonomous Flag    : On
   Advertise Flag     : On
```

```
IPv6 Prefix         : 2001:db8:f:1b::/64
Valid Lifetime      : 11/31/2014 00:00:01
Preferred Lifetime  : 11/01/2014 00:00:01
On-link Flag        : Off
Autonomous Flag     : On
Advertise Flag      : On

IPv6 Prefix         : 2001:db8:f:1d::/64
Valid Lifetime      : 11/31/2014 00:00:01
Preferred Lifetime  : 11/01/2014 00:00:01
On-link Flag        : On
Autonomous Flag     : On
Advertise Flag      : On
```

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 (DoS) 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 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>]
```

Syntax:

```
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:

```
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 IPv6 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, see **DNS resolver for IPv6** on page 137.

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, see the current *ArubaOS-Switch Management and Configuration Guide* for your switch.

Syntax:

```
ping6 <ipv6-address|hostname|switch-number> [repetitions <1-10000>] [timeout <1-
60>] [data-size <0-65471>] [data-fill <0-1024>] [source <ipv6-addr|vid>]
```

Syntax:

```
ping6 <link-local-address> %vlan <vid>|hostname|switch-number> [repetitions <1-
10000>] [timeout <1-60>] [data-size <0-65471>] [data-fill <0-1024>] [source <ipv6-
addr|vid>
```

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 <vid>` : IPv6 link-local address, where `%vlan <vid>` specifies the VLAN ID number.

`<hostname>` : Host name of an IPv6 host device configured on an IPv6 DNS server.

`[repetitions <1-10000>]` : Number of times that IPv6 ping packets are sent to the destination IPv6 host. Default: 1.

`[timeout <1-60>]` : 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.

`[data-size <0-65471>]` : Size of data (in bytes) to be sent in ping packets. Valid values: 0–65471. Default: 0.

`[data-fill <0-1024>]` : Text string used as data in ping packets. Default: 0; Range: up to 1024 alphanumeric characters.

`source [<ipv6-addr|vid>]` : The IPv6 address of the pinging device or the VLAN–ID on which the ping is being sent. Default : 0 (no text is used).

IPv6 ping tests

```
Switch# ping6 fe80::2:1%vlan10
fe80:0000:0000:0000:0000:0000:0002:0001 is alive, time = 975 ms

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

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

Switch# ping6 2001:db8::10
Request timed out.
```

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, see **DNS resolver for IPv6** on page 137.

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, see the *ArubaOS-Switch Management and Configuration Guide* for your switch.

Syntax:

```
traceroute6 <ipv6-address|hostname> [minttl <1-255> [maxttl <1-255> [timeout <1-
120>] [probes <1-5>] [source <ipv6-addr|vid|loopback <0-7>] [dstport <1-34000>]
[srcport <1-34000>]
```

Syntax:

```
traceroute6 <link-local-address %vlan <vid>|hostname> [minttl <1-255> [maxttl <1-
255> [timeout <1-120>] [probes <1-5>] [source <ipv6-addr|vid>]
```

Lists 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: 1–120

`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|vid>]` : The source IPv6 address or VLAN of the traceroute device or the VLAN-ID on which the traceroute packet is being sent.

`[dstport <1-34000>]` : Destination port.

`[srcport <1-34000>]` : Source port.

IPv6 traceroute probes

```
Switch# traceroute6 2001:db8::10
traceroute to 2001:db8::10
      1 hop min, 30 hops max, 5 sec. timeout, 3 probes
 1 2001:db8::a:1c:e3:3           0 ms    0 ms    0 ms
 2 2001:db8:0:7::5               7 ms    3 ms    0 ms
 3 2001:db8::214:c2ff:fe4c:e480  0 ms    1 ms    0 ms
 4 2001:db8::10                  0 ms    1 ms    0 ms
```

First three hops: Intermediate router hops with the time (in milliseconds) for the switch to receive a response from each of the three probes sent to each router.

Last hop: Destination IPv6 address

```
Switch# traceroute6 2001:db8::10 maxttl 7
traceroute to fe80::1:2:3:4
      1 hop min, 7 hops max, 5 sec. timeout, 3 probes
1 2001:db8::a:1c:e3:3           0 ms    0 ms    0 ms
2 2001:db8:0:7::5               0 ms    0 ms    0 ms
3 * 2001:db8::214:c2ff:fe4c:e480 *
4 * * *
5 * * *
6 * * *
7 * * *
```

At hop 3, the first and third probes timed out, but the second probe reached the router. Each timed-out probe is displayed with an asterisk (*).

The four remaining probes within the configured seven-hop maximum (`maxttl`) also timed out without finding a next-hop router or the destination IPv6 address.

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

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 hostname of the desired target when executing a command that supports DNS operation. But if the DNS server used to resolve the hostname 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.

NOTE:

This section describes the commands for configuring DNS operation for IPv6 DNS applications. For further information and examples on using the DNS feature, see “DNS Resolver” in appendix, “Troubleshooting”, in the current *Management and Configuration Guide* for your switch.

ip dns server-address priority

Syntax

```
ip dns server-address priority <PRIORITY> <IP-ADDR | IPv6-ADDR> [ oobm ]

no ip dns server-address priority <PRIORITY> <IP-ADDR | IPv6-ADDR> [ oobm ]
```

Description

Configures the IP address and priority of the DNS server. The command allows both IPv4 and IPv6 servers in any combination and any order of priority. The `ip dns server-address priority` supports a maximum of four manual DNS servers. The priorities are 1, 2, 3 and 4.

The `no` form of the command removes the specified IP address from the server address list configured on the switch.

NOTE: Windows DHCPv6 server is not supported.

Command context

`config`

Parameters

`priority <1-4>`

Specifies the priority of the DNS address.

- Priority 1 and 2 are reserved for IPv4 address by default when configured through DHCP server.
- Priority 3 and 4 are reserved for IPv6 address by default when configured through DHCP server.
- Configure the IPv4 or IPv6 address manually to set any priorities.

IP-ADDR | IPv6-ADDR

Specifies the address of an IPv6 or IPv4 DNS server.

oobm

Specifies that the communication with the DNS server goes through the OOBM port.

Example

```
Switch(config)# ip dns
 dhcp                  Configure DNS via DHCP.
 domain-name           Configure the DNS (Domain Name System) domain name for translation of
                       hostnames to IP addresses.
 server-address        Configure DNS server IP address.
Switch(config)# ip dns server-address
 priority              Priority of Server Address.
Switch(config)# ip dns server-address priority
 <1-4>                 Enter a number.
Switch(config)# ip dns server-address priority 4
 IPV6-ADDR             DNS server IPv6 address.
 IP-ADDR               DNS server IP address.
Switch(config)# ip dns server-address priority 4 2001:1db8:3cd4:1115:1111:2222:1a2f:1a2b
 oobm                  Use the OOBM interface to connect to the server.

Switch(config)# ip dns server-address priority 4 2001:1db8:3cd4:1115:1111:2222:1a2f:1a2b oobm
```

show ip dns

```
Switch(config)# show ip dns

  DNS Mode               :   Manual

  DNS Domain Names Configured:
  ---------------------------

  No.       DNS Domain Name
  ---       ---------------
  1.        dmn3


  DNS Servers Configured:
  ----------------------

  Priority DNS Server Addresses                     OOBM
  -------- ---------------------------------------- ----
    1.     2001:1db8:3cd4:1115:1111:2222:1a2f:1a1b  YES
    3.     192.168.1.1                              YES
    4.     2001:1db8:3cd4:1115:1111:2222:1a2f:1a2b  NO
```

show running-configuration

```
; JL071A Configuration Editor; Created on release #KB.16.06.0000x
; Ver #13:03.f8.1c.fb.7f.bf.bb.ff.7c.59.fc.7b.ff.ff.fc.ff.ff.3f.ef:05

hostname "Aruba-3810M-24G-1-slot"
module 1 type jl071x
flexible-module A type JL081A
ip dns domain-name "dmn3"
```

```
ip dns server-address priority 1 2001:1db8:3cd4:1115:1111:2222:1a2f:1a1b oobm
ip dns server-address priority 3 192.168.1.1 oobm
ip dns server-address priority 4 2001:1db8:3cd4:1115:1111:2222:1a2f:1a2b
```

ip dns domain-name

Syntax

`ip dns domain-name` *`domain-name-suffix`*

`no ip dns domain-name` *`domain-name-suffix`*

Description

Used at the global config level to configure the domain suffix that is automatically appended to the hostname 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 hostnames. The `no` form of the command removes the configured domain name suffix.

Example input

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.hpnetworking.net
- a priority of 1 for the above server
- the domain suffix mygroup.hpnetworking.net

Assume that the above, configured DNS server supports an IPv6 device having a hostname of “mars-1” (and an IPv6 address of fe80::215:60ff:fe7a:adc0) in the “mygroup.hpnetworking.net” domain. In this case you can use the device's hostname alone to ping the device because the mygroup.hpnetworking.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:

```
Switch(config)# ip dns server priority 1 2001:db8::127:10
Switch(config)# ip dns domain-name mygroup.hpnetworking.net
Switch(config)# ping6 mars-1
fe80::215:60ff:fe7a:adc0 is alive, time = 1 ms
```

However, for the same “mars-1” device, if mygroup.hpnetworking.net was not the configured domain name, you would have to use the fully qualified domain name for the device named mars-1:

```
Switch# ping6 mars-1.mygroup.hpnetworking.net
```

For further information and examples on using the DNS feature, see “DNS Resolver” in the Troubleshooting appendix, in the current *Management and Configuration Guide* for your switch.

Viewing the current DNS 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.

Debug/Syslog for IPv6

The Debug/System logging (Syslog) for IPv6 feature provides logging functions similar to those of 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 mis-configurations and other network protocol details to an external device, and later use them to debug network-level problems.

NOTE:

This section describes the commands for Debug/Syslog configuration in an IPv6 environment. For information on using the Debug/Syslog feature in an IPv4 environment, see “Debug/Syslog Operation” in the current management and configuration guide for your switch.

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 ipv6` command to send messaging reports for the following types of switch events:
  - DHCPv6 client
  - DHCPv6 relay
  - forwarding
  - neighbor discovery
  - packets
- 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:

`debug ipv6 [debug-type]`

`no debug ipv6 [debug-type]`

Configures the types of IPv6 messages that are sent to Syslog servers or other configured debug destinations, where debug-type is any of the following event types:

`(none)` : all IPv6 events

`dhcpv6-client <events|packets>` : one of the following IPv6 client debug message types

`events` : DHCPv6 client events

`packets` : DHCPv6 client packets

`dhcpv6-relay <events|packets>` : one of the following IPv6 relay debug message types

`events` : DHCPv6 relay events

`packets` : DHCPv6 relay packets

`forwarding` : IPv6 forwarding events

`nd` : IPv6 neighbor discovery events

`packet` : all IPv6 packet messages

The `no debug ipv6` form of the command stops the sending of debug messages of the specified 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.

`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

An IPv6-based 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 140).`debug destination logging` : enables the configured debug message types to be sent to Syslog servers configured with the `logging <syslog-ipv4addr|syslog-ipv6-addr>` command.`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.
- Use the `logging <syslog-ipv6-addr>` command to configure the Syslog server at the specified IPv6 destination address.

Logging command

Syntax:

`logging <syslog-ipv4-address>`

`no logging <syslog-ipv4-address>`

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, see the *ArubaOS-Switch Management and Configuration Guide*.

Networking Websites

Hewlett Packard Enterprise Networking Information Library

www.hpe.com/networking/resourcefinder

Hewlett Packard Enterprise Networking Software

www.hpe.com/networking/software

Hewlett Packard Enterprise Networking website

www.hpe.com/info/networking

Hewlett Packard Enterprise My Networking website

www.hpe.com/networking/support

Hewlett Packard Enterprise My Networking Portal

www.hpe.com/networking/mynetworking

Hewlett Packard Enterprise Networking Warranty

www.hpe.com/networking/warranty

General websites

Hewlett Packard Enterprise Information Library

www.hpe.com/info/EIL

For additional websites, see **Support and other resources**.

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  http://www.hpe.com/assistance
- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  http://www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates:

  Hewlett Packard Enterprise Support Center

  www.hpe.com/support/hpesc

  Hewlett Packard Enterprise Support Center: Software downloads

  www.hpe.com/support/downloads

  Software Depot

  www.hpe.com/support/softwaredepot
- To subscribe to eNewsletters and alerts:

  www.hpe.com/support/e-updates
- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

  www.hpe.com/support/AccessToSupportMaterials

IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

http://www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

Remote support and Proactive Care information

HPE Get Connected

www.hpe.com/services/getconnected

HPE Proactive Care services

www.hpe.com/services/proactivecare

HPE Proactive Care service: Supported products list

www.hpe.com/services/proactivecaresupportedproducts

HPE Proactive Care advanced service: Supported products list

www.hpe.com/services/proactivecareadvancedsupportedproducts

Proactive Care customer information

Proactive Care central

www.hpe.com/services/proactivecarecentral

Proactive Care service activation

www.hpe.com/services/proactivecarecentralgetstarted

Warranty information

To view the warranty information for your product, see the links provided below:

HPE ProLiant and IA-32 Servers and Options

www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise and Cloudline Servers

www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

www.hpe.com/support/Storage-Warranties

HPE Networking Products

www.hpe.com/support/Networking-Warranties

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

www.hpe.com/info/reach

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

www.hpe.com/info/ecodata

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

www.hpe.com/info/environment

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (**docsfeedback@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.

The following section is added in **Chapter 4 IPv6 Management Security Features** for the switch software version 16.08.0021. For more information see, Chapter 4 IPv6 Management Security Features.

SSH for IPv6

Beginning with software release K.14.01, SSH for IPv4 and IPv6 operate simultaneously with the same command set. Both are enabled in the default configuration, and are controlled together by the same command set. SSH for IPv6 provides the same Telnet-like functions through encrypted, authenticated transactions as SSH for IPv4. SSH for IPv6 provides CLI (console) access and secure file transfer functionality. The following types of transactions are supported:

- Client public-key authenticationPublic keys from SSH clients are stored on the switch. Access to the switch is granted only to a client whose private key matches a stored public key.
- Password-only client authenticationThe switch is SSH-enabled but is not configured with the login method that authenticates a client's public-key. Instead, after the switch authenticates itself to a client, users connected to the client authenticate themselves to the switch by providing a valid password that matches the operator- and/or manager-level password configured and stored locally on the switch or on a RADIUS or TACACS+ server.
- Secure Copy (SCP) and Secure File Transfer Protocol (SFTP) client applicationsYou can use either one SCP session or one SFTP session at a given time to perform secure file transfers to and from the switch.

By default, SSH is automatically enabled for IPv4 and IPv6 connections on a switch. Use the `ip ssh` command options to recon igure the default SSH settings used in SSH authentication for IPv4 and IPv6 connections:

- TCP port number
- timeout period
- ile transfer
- MAC type
- cipher type
- listening port

ip ssh

Syntax

```
ip ssh
```

```
no ip ssh
```

Description

Enables SSH for on the switch for both IPv4 and IPv6, and activates the connection with a con igured SSH server. The `no` form of the command disables SSH on the switch.

Parameters

`rekey {time <`*`time`*`> | volume <`*`volume`*`>}`

Enable SSH key re-exchange.

To comply with RFC 4251, session rekeying ensures that either the SSH server or the SSH client initiates a rekey. This results in a new set of encryption and integrity keys to be exchanged between them. Once the rekey is complete, new keys are used for further communication, which ensures that the same key is not used for a long duration and the security of the session is maintained.

NOTE: SSH rekeying is available on switches running KB, WC, WB, and YC softwares.

Valid options are:

- `time <time>`

  Sets the time in minutes for rekey initiation; the range is 10 to 60.

- `volume <volume>`

  Sets the volume in KB for rekey initiation; the range is 100-1048576. The default is 1048576 KB.

Use the `no` form of the command to disable SSH rekeying and set the time to default value of 10 minutes.

`cipher cipher-type`

Specifies a cipher type to be used for the connection.

Valid types are:

- aes128–cbc
- 3des-cbc
- aes192–cbc
- aes256–cbc
- rijndael-cbc@lysator.liu.se
- aes128–ctr
- aes192–ctr
- aes256–ctr

Default: All cipher types are available.

Use the `no` form of the command to disable a cipher type.

`filetransfer`

Enables SSH on the switch to connect to an SCP or SFTP client application to transfer iles to and from the switch over IPv4 or IPv6.

Default: Disabled

NOTE: Enabling `filetransfer` automatically disables TFTP client and TFTP server functionality.

`kex`

Enables SSH Key Exchange (KEX) algorithms on the switch connection with a configured SSH server. Valid types include:

- ecdh-sha2-nistp256
- ecdh-sha2-nistp384
- ecdh-sha2-nistp521
- diffie-hellman-group-exchange-sha256
- diffie-hellman-group14-sha1

Default: All key exchange algorithms are enabled.

Use the `no` form of the command to disable SSH KEX algorithms on the switch.

`mac <MAC-type>`

Enables type of Message Authentication Code (MAC) to be used. Valid MAC types include:

- hmac-md5
- hmac-sha1
- hmac-sha1–96
- hmac-md5–96

Default: All MAC types are available.

Use the `no` form of the command to disable a MAC type.

`port [1 - 65535|default]`

TCP port number used for SSH sessions in IPv4 and IPv6 connections.

Default: 22

Valid port numbers are from 1 to 65535, except for port numbers 23, 49, 80, 280, 443, 1506, 1513, and 9999, which are reserved for other subsystems.

`public-key [manager|operator]keystring`

Stores a client-generated key for public-key authentication.

manager

Allows manager-level access using SSH public-key authentication.

operator

Allows operator-level access using SSH public-key authentication.

keystring

A legal SSHv2 (RSA or DSA) public key. The text string for the public key must be a single-quoted token. If the keystring contains double quotes, it can be quoted with single quotes ('key-string'). The following restrictions are applicable for a keystring:

- A keystring cannot contain both single and double quotes.
- A keystring cannot have extra characters, such as a blank space or a new line. (To improve readability, you can add a backslash at the end of each line.)

`timeout 5 - 120`

Time out value allowed to complete an SSH authentication and login on the switch.

Default: 120 seconds.

`listen [oobm|data|both]`

The `listen` parameter is available only on switches that have a separate OOBM port. Values for this parameter are:

oobm

Inbound SSH access is enabled only on the OOBM port.

data

Inbound SSH access is enabled only on the data ports.

both

Inbound SSH access is enabled on both the OOBM port and on the data ports. This is the default value.

Restrictions

For both IPv4 and IPv6, the switch supports only SSH version 2. You cannot set up an SSH session with a client device running SSH version 1.

The `listen` parameter is not available on switches that do not have a separate OOBM port.

Examples

Use the following command to initiate rekeying every 45 minutes

```
switch(config)# ip ssh rekey time 45
```

Initiate rekeying every 45 minutes.

Use the following command to reset the rekey time

```
switch(config)# no ip ssh rekey time
```

Reset the configured time to the default value (60 minutes).

Use the following command to initiate rekeying after a speci ic data transfer

```
switch(config)# ip ssh rekey volume 2000
```

Initiate rekeying after every 2000 KB of data is transferred.

Use the following command to reset the data transfer volume

```
switch(config)# no ip ssh rekey volume
```

Reset the con igured volume to the default value (1048576 KB).

show ip ssh

Syntax

```
show ip ssh
```

Description

To verify an SSH con iguration and display all SSH sessions running on the switch, enter the `show ip ssh` command. Information on all current SSH sessions (IPv4 and IPv6) is displayed.

Restrictions

With SSH running, the switch supports one console session and up to ive other SSH and Telnet (IPv4 and IPv6) sessions. WebAgent sessions are also supported, but are not displayed in `show ip ssh` output.

Example

```
Switch# show ip ssh
Enabled          : Yes Secure Copy Enabled : No
TCP Port Number : 22   Timeout (sec)        : 120
Rekey Enabled   : No   Rekey time (min)     : 60
                  Rekey Volume (KB)         : 1048576
 Host Key Type   : RSA           Hot Key/Curve Key  : 2048
Ciphers : aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc,
          rijndael-cbc@lysator.liu.se,aes128-ctr,aes192-ctr,
          aes256-ctr
MACs    : hmac-md5,hmac-sha1,hmac-sha1-96,hmac-
Kex     : ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sh
          diffie-hellman-group14-sha1

Ses Type     | Source IP                                Port
--- -------- + ---------------------------------------- -----
1   console  |
2   ssh      | 10.168.31.114                             1722
3   inactive |
4   inactive |
5   inactive |
6   inactive |
```