

AOS-CX 10.09 ACLs and Classifier Policies Guide

8400 Switch Series



a Hewlett Packard
Enterprise company

Copyright Information

© Copyright 2022 Hewlett Packard Enterprise Development LP.

Open Source Code

This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett Packard Enterprise Company. To obtain such source code, send a check or money order in the amount of US \$10.00 to:

Hewlett Packard Enterprise Company
6280 America Center Drive
San Jose, CA 95002
USA

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.

Contents	3
About this document	5
Applicable products	5
Latest version available online	5
Command syntax notation conventions	5
About the examples	6
Identifying switch ports and interfaces	6
Identifying modular switch components	7
Access Control Lists	8
ACL usage tips	9
About address and port object groups	10
ACL and ACE-related tasks	10
Object group-related tasks	12
Active ACL configuration versus user-specified configuration	13
ACL commands	15
ACL application	15
access-list log-timer	15
access-list copy	18
access-list ip	21
access-list ipv6	29
access-list mac	37
access-list resequence	43
access-list reset	45
apply access-list control-plane	48
apply access-list (to interface or LAG)	49
apply access-list (to interface VLAN)	51
apply access-list (to VLAN)	52
clear access-list hitcounts	54
clear access-list hitcounts control-plane	55
object-group address resequence	56
object-group address reset	56
object-group all reset	57
object-group ip address	58
object-group ipv6 address	60
object-group port	62
object-group port resequence	64
object-group port reset	65
show access-list	65
show access-list control-plane	70
show access-list hitcounts	72
show access-list hitcounts control-plane	75
show capacities	76
show capacities-status	78
show object-group	79
ACL configuration examples	82
IPv4 ACL example overview	82

Defining and applying an IPv4 ACL	82
IPv6 ACL example overview	83
Defining and applying an IPv6 ACL	83
Classifier policies	85
Traffic policing	85
Types of policy actions	86
How policy matching works	87
Active class configuration versus user-specified configuration	87
Active policy configuration versus user-specified configuration	88
Considerations for when a policy is applied per interface	89
Classifier policy commands	91
Classifier policy application	91
apply policy (config-if, config-lag-if, config-if-vlan, config-vlan)	91
class copy	95
class ip	96
class ipv6	104
class mac	110
class resequence	114
class reset	116
clear policy hitcounts	116
policy	118
policy copy	122
policy resequence	123
policy reset	124
show class	125
show policy	126
Classifier policies configuration example	132
Configuring the classifier policies example	132
ACL and Policy hardware resource considerations	135
TCAM lookups	135
Matching precedence order	135
L4 port ranges	136
ACL and Policy hardware resource commands	137
show resources	137
Configurable high capacity TCAM	139
Configuring the high capacity TCAM	139
Performing the configuration	139
Verifying the updated configuration	140
VSX sync requirements	140
High capacity TCAM commands	141
system high-capacity-tcam access-list ip	141
show system high-capacity-tcam	143
Support and Other Resources	145
Accessing Aruba Support	145
Accessing Updates	146
Aruba Support Portal	146
My Networking	146
Warranty Information	146
Regulatory Information	146
Documentation Feedback	147

This document describes features of the AOS-CX network operating system. It is intended for administrators responsible for installing, configuring, and managing Aruba switches on a network.

Applicable products

This document applies to the following products:

- Aruba 8400 Switch Series (JL375A, JL376A)

Latest version available online

Updates to this document can occur after initial publication. For the latest versions of product documentation, see the links provided in [Support and Other Resources](#).

Command syntax notation conventions

Convention	Usage
<code>example-text</code>	Identifies commands and their options and operands, code examples, filenames, pathnames, and output displayed in a command window. Items that appear like the example text in the previous column are to be entered exactly as shown and are required unless enclosed in brackets ([]).
example-text	In code and screen examples, indicates text entered by a user.
Any of the following: <ul style="list-style-type: none">■ <code><example-text></code>■ <code><example-text></code>■ <i>example-text</i>■ <i>example-text</i>	Identifies a placeholder—such as a parameter or a variable—that you must substitute with an actual value in a command or in code: <ul style="list-style-type: none">■ For output formats where italic text cannot be displayed, variables are enclosed in angle brackets (< >). Substitute the text—including the enclosing angle brackets—with an actual value.■ For output formats where italic text can be displayed, variables might or might not be enclosed in angle brackets. Substitute the text including the enclosing angle brackets, if any, with an actual value.
	Vertical bar. A logical OR that separates multiple items from which you can choose only one. Any spaces that are on either side of the vertical bar are included for readability and are not a required part of the command syntax.
{ }	Braces. Indicates that at least one of the enclosed items is required.
[]	Brackets. Indicates that the enclosed item or items are optional.

Convention	Usage
... or . . .	Ellipsis: <ul style="list-style-type: none"> ■ In code and screen examples, a vertical or horizontal ellipsis indicates an omission of information. ■ In syntax using brackets and braces, an ellipsis indicates items that can be repeated. When an item followed by ellipses is enclosed in brackets, zero or more items can be specified.

About the examples

Examples in this document are representative and might not match your particular switch or environment. The slot and port numbers in this document are for illustration only and might be unavailable on your switch.

Understanding the CLI prompts

When illustrating the prompts in the command line interface (CLI), this document uses the generic term `switch`, instead of the host name of the switch. For example:

```
switch>
```

The CLI prompt indicates the current command context. For example:

```
switch>
```

Indicates the operator command context.

```
switch#
```

Indicates the manager command context.

```
switch (CONTEXT-NAME)#
```

Indicates the configuration context for a feature. For example:

```
switch(config-if) #
```

Identifies the `interface` context.

Variable information in CLI prompts

In certain configuration contexts, the prompt may include variable information. For example, when in the VLAN configuration context, a VLAN number appears in the prompt:

```
switch(config-vlan-100) #
```

When referring to this context, this document uses the syntax:

```
switch(config-vlan-<VLAN-ID>) #
```

Where `<VLAN-ID>` is a variable representing the VLAN number.

Identifying switch ports and interfaces

Physical ports on the switch and their corresponding logical software interfaces are identified using the format:

```
member/slot/port
```

On the 8400 Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Specifies physical location of a module in the switch chassis.
 - Management modules are on the front of the switch in slots 1/5 and 1/6.
 - Line modules are on the front of the switch in slots 1/1 through 1/4, and 1/7 through 1/10.
- *port*: Physical number of a port on a line module

For example, the logical interface 1/1/4 in software is associated with physical port 4 in slot 1 on member 1.

Identifying modular switch components

- Power supplies are on the front of the switch behind the bezel above the management modules. Power supplies are labeled in software in the format: *member/power supply*:
 - *member*: 1.
 - *power supply*: 1 to 4.
- Fans are on the rear of the switch and are labeled in software as: *member/tray/fan*:
 - *member*: 1.
 - *tray*: 1 to 4.
 - *fan*: 1 to 4.
- Fabric modules are not labeled on the switch but are labeled in software in the format: *member/module*:
 - *member*: 1.
 - *member*: 1 or 2.
- The display module on the rear of the switch is not labeled with a member or slot number.

Access Control Lists (ACLs) let a network administrator permit or deny passage of traffic based on network addresses, protocols, service ports, and other packet attributes. ACLs are composed of one or more Access Control Entries (called ACEs). Each ACE defines a filter criteria and an action, either **permit** or **deny**. If the traffic matches the filter criteria, the specified action is taken. The **permit** action permits the traffic to continue through the switch. The **deny** action causes the traffic to be discarded (dropped). ACEs can also log or count matching traffic.

Three ACL types are supported; IPv4, IPv6, and MAC. Each ACL type is focused on relevant frame or packet characteristics.

ACLs must be applied (using an `apply access-list` command) to take effect. ACLs can be applied to interfaces (including LAGs), VLANs, or the Control Plane.

Access Control Entries (ACEs) are listed according to priority by sequence number and processed in lowest to highest sequence number order. Each ACE attempts to match on one or more attributes of the particular traffic type. Attempted ACE matching ceases upon the first successful match. For a match to be considered successful, a packet must match all the criteria, qualifiers, and attributes of a particular ACE. Higher-numbered ACEs are only processed if no lower-numbered ACE matches. If the traffic matches no ACE in the entire ACL, the default action **deny** is taken, causing the traffic to be discarded (dropped).

When defining an ACE, if the sequence number is omitted, the ACE is auto-assigned a new sequence number that is 10 greater than the existing highest ACE sequence number. The first auto-assigned sequence number is 10. If you choose to include the ACE sequence numbers, you can use any number you like, however it is suggested that you follow the practice of entering them as 10, 20, 30, and so on. Regardless of the order in which ACEs are entered, they are stored in low-to-high sequence number order. If you enter three ACEs numbered 10, 30, 20, when creating an ACL, the ACEs are stored in the ACL as 10, 20, 30.

This simple ACL definition permits traffic passage for a particular address range and otherwise counts all nonmatching (dropped) traffic:

```
switch(config)# access-list ip network-A-udp-only
switch(config-acl-ip)# 10 permit udp any 172.16.1.0/24
switch(config-acl-ip)# 20 deny any any any count
switch(config-acl-ip)# exit
```

The main traffic characteristics that ACEs can filter on are as follows (see the full list in the ACE parameters list of the ACL commands):

- Protocol such as: ICMP, TCP, UDP
- Source and/or destination addresses (IPv4, IPv6, or MAC)
- Source and/or destination TCP/UDP ports (if applicable to the specified protocol)

A few real-world uses of ACLs are as follows:

- Restrict traffic arriving on a routed port, destined to a particular address or subnet by applying an ACL that matches on a destination IP address or an IP address and a mask.
- Prevent an entire subnet from routing through a port by applying an ACL that matches on IP source address and a mask.

- Prevent certain protocols from using a particular multicast MAC address (advertising through a port) by applying an ACL that matches on the destination MAC address.
- Prevent any IP host from accessing a particular IP port/application on a specific server by applying an ACL that matches on IP addresses and Layer 4 port.



See also [ACL and Policy hardware resource considerations](#).

ACL usage tips

When using the `access-list ip` or `access-list ipv6` commands, if you enter an existing `ACL-NAME`, the existing ACL is modified as follows:

- Any ACE entered with a new sequence-number creates an additional ACE.
- Any ACE entered with an existing sequence-number replaces the existing ACE.

If you modify an ACL that has already been applied, it is possible that packets, blocked by the previous ACL, will briefly pass through the switch during the ACL reconfiguration.



In a highly secure environment, it is safest to first bring down interfaces and VLANs to which an ACL has been applied before modifying the ACL. Then bring the targets of ACL application back up after completing the ACL modification. Respecting this recommendation ensures that an ACL is never partially programmed while traffic is passing through the switch.

About applying ACLs to interfaces or LAGs

You can apply an ACL to an interface or LAG to affect or control the traffic arriving on that interface or LAG (inbound) or leaving the interface or LAG (outbound), or both. A given interface or LAG supports the application of a single ACL per type, per direction. ACLs can be applied to interfaces or LAGs as follows:

- One MAC ACL inbound
- One IPv4 ACL inbound
- One IPv4 ACL outbound
- One IPv6 ACL inbound

Different ACLs of the same type can be used in opposite directions for IPv4. If you apply an ACL of a particular type, in a direction that is already in use, the switch replaces the current ACL with the new ACL.

About applying ACLs to VLANs

ACLs can be applied to VLANs only in the inbound (ingress) direction.

Sequence numbering

If no sequence number is specified, the software appends new ACEs to the end of the ACL with a sequence number equal to the highest ACE currently in the list plus 10.

The sequence numbers may be resequenced using the `access-list resequence` command.

Deny ACLs

If multiple ACLs of different types are applied in the same direction, a deny ACE, whether explicit or implicit, in one ACL overrides a permit ACL in another. A deny ACE is an ACE within an ACL that uses the `deny` action keyword.

Denied ping requests

A ping request is denied when an ACL is applied on ingress or egress unless the request is explicitly permitted.

```
switch# ping 100.1.2.10
PING 100.1.2.10 (100.1.2.10) 100(128) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
```

About address and port object groups

Object groups are useful for defining groups of IP addresses and Layer 4 ports for use exclusively in the two ACL-defining commands `access-list ip` and `access-list ipv6`.

Often, common groups of addresses and ports or port ranges are used repeatedly in many ACL definitions. Without address and port object groups, the same addresses and ports must be repeated in each ACL definition that uses them.

With address and port object groups, the IP addresses and ports can be defined once, using any of these commands:

- `object-group ip address`
- `object-group ipv6 address`
- `object-group port`

Once an object group is defined, the group is available for inclusion by name as the `<ADDRESS-GROUP>` and `<PORT-GROUP>` parameters in the `access-list ip` and `access-list ipv6` ACL-definition commands.

Object groups simplify the ACL definition process and help ensure consistent address and port specification across many ACLs.



Keep in mind that it is possible to consume many hardware resource entries when using the object group commands. For example, in a typical situation, an ACE that uses object groups with 3 source addresses, 3 source L4 ports, 3 destination addresses, and 3 destination L4 ports, a total of 81 hardware entries are consumed ($3 * 3 * 3 * 3 = 81$).

ACL and ACE-related tasks

Common ACL and ACE-related tasks are as follows:

Task	Command name	Example
Creating an IPv4 ACL	<code>access-list ip</code>	<pre>access-list ip MY_IP_ACL 10 permit udp any 172.16.1.0/24 20 permit tcp 172.16.2.0/16 gt 1023 any 30 deny any any any count</pre>

Task	Command name	Example
Creating an IPv6 ACL	<code>access-list ipv6</code>	<code>access-list ipv6 MY_IPV6_ACL 10 permit udp any 2001::1/64 20 permit tcp 2001:2011::1/64 any 30 deny any any any count</code>
Creating a MAC ACL	<code>access-list mac</code>	<code>access-list mac MY_MAC_ACL 10 permit any any appletalk vlan 40 20 deny any any any count</code>
Applying an IPv6 ACL to an interface	<code>apply access-list (to interface or LAG)</code>	<code>interface 1/1/1 apply access-list ipv6 MY_IPV6_ACL in</code>
Applying an IPv4 ACL to a LAG	<code>apply access-list (to interface or LAG)</code>	<code>interface lag 100 apply access-list ip MY_IP_ACL in</code>
Applying an IPv4 ACL to a VLAN	<code>apply access-list (to VLAN)</code>	<code>vlan 10 apply access-list ip MY_IP_ACL in</code>
Applying a MAC ACL to a VLAN	<code>apply access-list (to VLAN)</code>	<code>vlan 40 apply access-list mac MY_MAC_ACL in</code>
Applying an IPv4 ACL to the Control Plane (OOBM)	<code>apply access-list control-plane</code>	<code>apply access-list ip MY_IP_ACL control-plane vrf mgmt</code>
Removing application of an ACL from an interface	<code>apply access-list (to interface or LAG)</code>	<code>interface 1/1/1 no apply access-list ipv6 MY_IPV6_ACL in</code>
Removing application of an ACL from a VLAN	<code>apply access-list (to VLAN)</code>	<code>vlan 40 no apply access-list mac MY_MAC_ACL in</code>
Removing application of an ACL from the Control Plane (OOBM)	<code>apply access-list control-plane</code>	<code>no apply access-list ip MY_IP_ACL control-plane vrf mgmt</code>
Showing all ACLs	<code>show access-list</code>	<code>show access-list</code>
Showing all IPv6 ACLs	<code>show access-list</code>	<code>show access-list ipv6</code>
Showing all ACLs applied to interface 1/1/1	<code>show access-list</code>	<code>show access-list interface 1/1/1</code>
Showing all ACLs applied to VLAN 10	<code>show access-list</code>	<code>show access-list vlan 10</code>
Showing all ACLs applied to the Control Plane	<code>show access-list control-plane</code>	<code>show access-list control-plane</code>
Showing a particular ACL	<code>show access-list</code>	<code>show access-list ip MY_ACL</code>
Showing an ACL as commands	<code>show access-list</code>	<code>show access-list ip MY_ACL commands</code>
Showing ACL hit counts for an ACL applied to an interface	<code>show access-list hitcounts</code>	<code>show access-list hitcounts ip MY_ACL interface 1/1/1</code>

Task	Command name	Example
Showing ACL hit counts for an ACL applied to a VLAN	<code>show access-list hitcounts</code>	<code>show access-list hitcounts ip MY_ACL vlan 10</code>
Showing ACL hit counts for an ACL applied to the Control Plane	<code>show access-list hitcounts control-plane</code>	<code>show access-list hitcounts ip MY_ACL control-plane vrf mgmt</code>
Clearing ACL hit counts	<code>clear access-list hitcounts</code>	<code>clear access-list hitcounts ip MY_ACL vlan 10</code>
Clearing ACL hit counts for Control Plane	<code>clear access-list hitcounts control-plane</code>	<code>clear access-list hitcounts control-plane vrf mgmt</code>
Copying an ACL	<code>access-list copy</code>	<code>access-list ipv6 MY_IPV6_ACL copy MY_IPV6_ACL2</code>
Resequencing the ACEs of an ACL	<code>access-list resequence</code>	<code>access-list ip MY_IP_ACL resequence 1 1</code>
Resetting an ACL	<code>access-list reset</code>	<code>access-list ip MY_IP_ACL reset</code>
Setting the ACL log timer frequency	<code>access-list log-timer</code>	<code>access-list log-timer 30</code>

Object group-related tasks

Object groups are useful for defining groups of addresses and ports for use exclusively in the two ACL-defining commands `access-list ip` and `access-list ipv6`.

Common object group-related tasks are as follows:

Task	Command name	Example
Creating an IPv4 address object group	<code>object-group ip address</code>	<code>object-group ip address my_ipv4_addr_group</code>
Creating an IPv6 address object group	<code>object-group ipv6 address</code>	<code>object-group ipv6 address my_ipv6_addr_group</code>
Creating a port object group	<code>object-group port</code>	<code>object-group port my_port_group</code>
Showing an IPv4 address object group	<code>show object-group</code>	<code>show object-group ip address my_ipv4_addr_group</code>
Showing all IPv6 address object groups	<code>show object-group</code>	<code>show object-group ipv6 address</code>

Task	Command name	Example
Showing a port object group	<code>show object-group</code>	<code>show object-group port my_port_group</code>
Showing all port object groups as commands	<code>show object-group</code>	<code>show object-group port commands</code>
Resequencing an IPv4 address object group	<code>object-group ip address</code>	<code>object-group ip address my_ipv4_addr_group resequence 100 10</code>
Resequencing a port object group	<code>object-group port</code>	<code>object-group port my_port_group resequence 200 5</code>
Resetting an IPv6 address object group	<code>object-group ipv6 address</code>	<code>object-group ipv6 address my_ipv6_addr_group reset</code>
Resetting a port object group	<code>object-group port</code>	<code>object-group port my_port_group reset</code>

Active ACL configuration versus user-specified configuration

The `show access-list` command shows the active configuration of the switch. The active configuration is the ACLs that have been configured and accepted by the system. The active configurations are the interfaces on which the ACLs have successfully been programmed in the hardware.

The output of the `show access-list` command with the `configuration` parameter shows the ACLs that have been configured. The output of this command may not be the same as what was programmed in the hardware or what is active on the switch. The situation might occur because of one or more of the following:

- Unsupported command parameters might have been configured.
- Unsupported applications might have been specified.
- Applying an ACL might have been unsuccessful due to lack of hardware resources.

To determine if a discrepancy exists between what was configured and what is active, run the `show access-list` command with the `configuration` parameter.

If the active ACLs and configured ACLs are not the same, the switch shows a warning message in the output of the `show` command:

```
! access-list ip MY_IP_ACL user configuration does not match active configuration.
! run 'access-list TYPE NAME reset' to reset access-list to match active configuration.
```

If the configured ACL is processing, the switch shows an in-progress warning.

```
! access-list ip MY_IP_ACL user configuration currently being processed
! run 'access-list TYPE NAME reset' to reset access-list to match active configuration.
```

If the switch shows a warning message or in-progress message, additional changes can be made until the error message is no longer shown in the show command, or you can run the `access-list {all|ip <ACL-NAME>|ipv6 <ACL-NAME>|mac <ACL-NAME>} reset` command. The `access-list reset` command changes the user-specified configuration to match the active configuration. For details, see [access-list reset](#).



The `show running-config` command also shows a warning about ACLs that are in progress or failed.

Examples

Applying an ACL with TCP acknowledgments (ACKs) on ingress, which is unsupported by the hardware:

```
switch(config-acl)# 10 permit tcp 172.16.2.0/16 any ack
```

Showing the user-specified configuration:

```
switch(config)# do show access-list ip TEST_ACL
  10 permit tcp 172.16.2.0/16 any ack
  interface 1/1/1
  ! access-list ip TEST_ACL user configuration does not match active
  configuration.
  ! run 'show access-list [commands]' to display active access-list configuration.
  apply access-list ip TEST_ACL in

switch(config)# do show access-list commands
access-list ip TEST_ACL
  10 permit tcp 172.16.2.0/16 any ack
! access-list ip TEST_ACL user configuration does not match active configuration.
! run 'access-list all reset' to reset all access-lists to match active
configuration.

switch(config)# do show access-list commands configuration
access-list ip TEST_ACL
  10 permit tcp 172.16.2.0/16 any ack
! access-list ip TEST_ACL user configuration does not match active configuration.
! run 'access-list all reset' to reset all access-lists to match active
configuration.
interface 1/1/1
  apply access-list ip TEST_ACL in

switch(config)# do show access-list
Type      Name
Sequence Comment
          Action          L3 Protocol
          Source IP Address Source L4 Port(s)
          Destination IP Address Destination L4 Port(s)
          Additional Parameters
-----
IPv4      TEST_ACL
          10 permit          tcp
          172.16.2.0/16
          any
          ack
```

Resetting the user-specified configuration to match the active configuration:

```
switch(config)# access-list all reset
```

Showing the updated user-specified configuration:

```
switch(config)# do show access-list commands configuration
access-list ip TEST_ACL
10 permit tcp 172.16.2.0/16 any ack
```

ACL commands

ACL application

ACLs can be applied as follows:

ACL type Direction	IPv4+6 In	IPv4 Out	MAC In
L2 interface (port)	Yes		Yes
L2 LAG	Yes		Yes
L3 interface (port)	Yes	Yes	Yes
L3 LAG	Yes	Yes	Yes
VLAN	Yes		Yes
Interface VLAN	Yes (routed)		
Management interface	Yes		
Control plane (per VRF)	Yes		

Egress ACLs can only be applied to L3 (route-only) interfaces. Applying an egress ACL to an L2 interface will result in an error. Only ingress ACLs (IPv4, IPv6, and MAC) can be applied to VLANs. Applying an egress ACL to VLAN will result in an error.

The following match criteria are not supported. If any of these match criteria are attempted to be configured, an error message will be displayed and the action will not be completed.



```
TCP flags CWR and ECE
TCP flags and TTL (hop limit) on IPv6 ACLs
TCP flags and fragment on outbound ACLs
Fragment on IPv6 VLAN ACLs
VLAN ID on VLAN ACLs
```

To apply IPv4 and/or IPv6 ACLs to the management interface, apply them to the control plane on the management VRF.

access-list log-timer

```
access-list log-timer {default|<INTERVAL>}
```

Description

Sets the log timer interval for all ACEs that have the `log` parameter configured.

Parameter	Description
default	Resets the log timer to its default 300 seconds.
<INTERVAL>	Specifies the log timer interval in seconds. Range: 5 to 300.

Usage

- ACL logging keeps a log of the number of packets matching this ACE. Works with `deny` actions but not with `permit` actions. Works with ACLs applied on ingress or egress, except for control plane.
- The first packet that matches an ACE with the `log` parameter within an ACL log timer window (configured with the `access-list log-timer` command) has its header contents extracted and sent to the configured logging destination, such as the console and syslog server. Each time the ACL log timer expires, a summary of all ACEs with `log` configured are sent to the logging destination. This capability allows throttling of logging ACL hits.
- If no further log messages are generated in the wait-period, the switch suspends the timer and resets itself to log as soon as a new match occurs.
- When using multiple ACL types (IPv4, IPv6, or MAC) with logging on the same interface, the first packet that matches an ACE with the `log` option is logged. Any packets, matching other ACL types, do not create a log until the log-timer wait-period is over. At the end of the wait-period, a summary log is made of all the ACLs that were matched, regardless of type.
- You may see a minor discrepancy between the ACL logging statistics and the hit counts statistics due to the time required to record the log message.

Examples



Although these examples use debug logging, you can alternatively use event logging.

Enabling debug logging for the ACL logging module:

```
switch# debug acl log severity info
switch# show debug
-----
module sub_module severity vlan port ip mac instance vrf
-----
acl acl_log info -----
```

Setting the debug destination to console with the minimum security level of info:

```
switch# debug destination console severity info
switch# show debug destination
-----
show debug destination
-----
CONSOLE:info
```

Setting the access list log-timer to 30 seconds:


```
switch(config)# access-list log-timer 30
switch(config)# do show access-list log-timer
ACL log timer length (frequency): 30 seconds
```

Creating an IPv4 ACL with one entry with the log parameter:

```
switch(config)# access-list ip MY_IP_ACL
switch(config-acl-ip)# deny icmp 1.1.1.1 1.1.1.2 log
switch(config-acl-ip)# do show access-list
Type          Name
Sequence      Comment
              Action                    L3 Protocol
              Source IP Address          Source L4 Port(s)
              Destination IP Address    Destination L4 Port(s)
              Additional Parameters
-----
IPv4          MY_IP_ACL
              10 deny                               icmp
              1.1.1.1
              1.1.1.2
              Logging: enabled
              Hit-counts: enabled
```

Enabling interface 1/1/1 and applying the ACL:

```
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# no routing
switch(config-if)# apply access-list ip MY_IP_ACL in
switch(config-if)# do show running-config interface 1/1/1
interface 1/1/1
  no shutdown
  apply access-list ip MY_IP_ACL in
  no routing
  vlan access 1
  exit
```

Sending packets that will match the ACE and observe the ACL logging message on the console:

```
2017-10-10T20:13:36.044+00:00 ops-switchd[875]: debug|LOG_INFO|AMM|1/5|ACL|ACL_LOG|
List MY_IP_ACL, seq# 10 denied icmp 1.1.1.1 -> 1.1.1.2 type 8 code 0,
on vlan 1, port 1/1/1, direction in
```

When the access list log-timer expires, the summary message is printed on the console. The number 30 is the number of packets received during the last access list log-timer window.

```
2017-10-10T20:14:06.051+00:00 ops-switchd[875]: debug|LOG_INFO|AMM|1/5|ACL|ACL_LOG|
MY_IP_ACL on 1/1/1 (in): 30 10 deny icmp 1.1.1.1 1.1.1.2 log count
```

Resetting the ACL log timer to the default value:

```
switch(config)# access-list log-timer default
```

Command History

Release	Modification
10.09	<INTERVAL> parameter range changed to 5 to 300 . Was 30 to 300.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

access-list copy

```
access-list {ip|ipv6|mac} <ACL-NAME> copy <DESTINATION-ACL>
```

Description

Copies an IPv4, IPv6, or MAC ACL to a new destination ACL or overwrites an existing ACL.

Parameter	Description
{ip ipv6 mac}	Specifies the type of ACL.
<ACL-NAME>	Specifies the name of the ACL to be copied.
<DESTINATION-ACL>	Specifies the name of the destination ACL.

Examples

Copying MY_IP_ACL to MY_IP_ACL2:

```
switch(config)# access-list ip MY_IP_ACL copy MY_IP_ACL2
switch(config-acl-ip)# exit

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action                L3 Protocol
          Source IP Address       Source L4 Port(s)
          Destination IP Address  Destination L4 Port(s)
          Additional Parameters
-----
IPv4      MY_IP_ACL
1 permit any 172.16.1.0/255.255.255.0 udp
2 permit 172.16.2.0/255.255.0.0 tcp > 1023
3 permit 172.26.1.0/255.255.255.0 tcp
          any
          dscp: AF11
          ack
```

```

        syn
    4 deny                                any
        any
        any
        Hit-counts: enabled
-----
IPv4    MY_IP_ACL2
    1 permit                                udp
        any
        172.16.1.0/255.255.255.0
    2 permit                                tcp
        172.16.2.0/255.255.0.0          > 1023
        any
    3 permit                                tcp
        172.26.1.0/255.255.255.0
        any
        dscp: AF11
        ack
        syn
    4 deny                                any
        any
        any
        Hit-counts: enabled

```

Copying MY_IPV6_ACL to MY_IPV6_ACL2:

```

switch(config)# access-list ipv6 MY_IPV6_ACL copy MY_IPV6_ACL2
switch(config-acl-ip)# exit

switch(config)# do show access-list
Type      Name
Sequence Comment
Action    L3 Protocol
Source IP Address Source L4 Port(s)
Destination IP Address Destination L4 Port(s)
Additional Parameters
-----
IPv6      MY_IPV6_ACL
    1 permit                                udp
        any
        2001::1/64
    2 Permit all TCP ephemeral ports
        permit                                tcp
        2001:2001::2:1                      > 1023
        any
    3 permit                                tcp
        2001:2011::1/64
        any
    4 deny                                any
        any
        any
        Hit-counts: enabled
-----
IPv6      MY_IPV6_ACL2
    1 permit                                udp
        any
        2001::1/64
    2 Permit all TCP ephemeral ports
        permit                                tcp
        2001:2001::2:1                      > 1023
        any

```

```

3 permit                               tcp
  2001:2011::1/64
  any
4 deny                                  any
  any
  any
Hit-counts: enabled

```

Copying MY_MAC_ACL to MY_MAC_ACL2:

```

switch(config)# access-list mac MY_MAC_ACL copy MY_MAC_ACL2
switch(config-acl-mac)# exit

```

```

switch(config)# do show access-list

```

```

Type          Name
Sequence      Comment
              Action                               EtherType
              Source MAC Address
              Destination MAC Address
              Additional Parameters
-----
MAC           MY_MAC_ACL
1 permit      1122.3344.5566/ffff.ffff.0000  ipv6
  any
2 permit      aaaa.bbbb.cccc                 any
  1111.2222.3333
  QoS Priority Code Point: 4
3 Permit all vlan-1 tagged Appletalk traffic
  permit      appletalk
  any
  any
  VLAN: 1
4 deny        any                               any
  any
  any
Hit-counts: enabled
-----
MAC           MY_MAC_ACL2
1 permit      1122.3344.5566/ffff.ffff.0000  ipv6
  any
2 permit      aaaa.bbbb.cccc                 any
  1111.2222.3333
  QoS Priority Code Point: 4
3 Permit all vlan-1 tagged Appletalk traffic
  permit      appletalk
  any
  any
  VLAN: 1
4 deny        any                               any
  any
  any
Hit-counts: enabled

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

access-list ip

Syntax to create an IPv4 ACL and enter its context. Plus syntax to remove an ACL:

```
access-list ip <ACL-NAME>
no access-list ip <ACL-NAME>
```

Syntax (within the ACL context) for creating or removing ACEs for protocols `ah`, `gre`, `esp`, `igmp`, `ospf`, `pim` (`ip` is available as an alias for `any`):

```
[<SEQUENCE-NUMBER>]
{permit|deny}
{any|ip|ah|gre|esp|igmp|ospf|pim|<IP-PROTOCOL-NUM>}
{any|<SRC-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}] |<ADDRESS-GROUP>}
{any|<DST-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}] |<ADDRESS-GROUP>}
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count] [log]

no <SEQUENCE-NUMBER>
```

Syntax (within the ACL context) for creating or removing ACEs for protocols `sctp`, `tcp`, `udp`:

```
[<SEQUENCE-NUMBER>]
{permit|deny}
{sctp|tcp|udp}
{any|<SRC-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}] |<ADDRESS-GROUP>}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>|group <PORT-GROUP>]
{any|<DST-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}] |<ADDRESS-GROUP>}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>|group <PORT-GROUP>]
[urg] [ack] [psh] [rst] [syn] [fin] [established]
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count] [log]

no <SEQUENCE-NUMBER>
```

Syntax (within the ACL context) for creating or removing ACEs for protocol `icmp`:

```
[<SEQUENCE-NUMBER>]
{permit|deny}
{icmp}
{any|<SRC-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}] |<ADDRESS-GROUP>}
{any|<DST-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}] |<ADDRESS-GROUP>}
[icmp-type {echo|echo-reply|<ICMP-TYPE-VALUE>}] [icmp-code <ICMP-CODE-VALUE>]
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count] [log]

no <SEQUENCE-NUMBER>
```

Syntax (within the ACL context) for ACE comments:

```
[<SEQUENCE-NUMBER>] comment <TEXT-STRING>

no <SEQUENCE-NUMBER> comment
```

Description

Creates an IPv4 Access Control List (ACL) comprised of one or more Access Control Entries (ACEs) ordered and prioritized by sequence number. The lowest sequence number is the highest prioritized ACE.

The `no` form of this command deletes the entire ACL, or deletes an ACE identified by sequence number, or deletes only the comment from the ACE identified by sequence number.

Parameter	Description
<ACL-NAME>	Specifies the name of this ACL.
<SEQUENCE-NUMBER>	Specifies a sequence number for the ACE. Range: 1 to 4294967295.
{permit deny}	Specifies whether to permit or deny traffic matching this ACE.
<IP-PROTOCOL-NUM>	Specifies the protocol as its Internet Protocol number. For example, 2 corresponds to the IGMP protocol. Range: 0 to 255.
{any <SRC-IP-ADDRESS>[/<PREFIX-LENGTH> <SUBNET-MASK>]} <ADDRESS-GROUP>	<p>Specifies the source IPv4 address.</p> <ul style="list-style-type: none"> ■ any - specifies any source IPv4 address. ■ <SRC-IP-ADDRESS> - specifies the source IPv4 host address. <ul style="list-style-type: none"> ○ <PREFIX-LENGTH> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 32. ○ <SUBNET-MASK> - specifies the address bits to mask (dotted decimal notation). ■ <ADDRESS-GROUP> - specifies an IPv4 address group defined with <code>object-group ip address</code>.
{any <DST-IP-ADDRESS>[/<PREFIX-LENGTH> <SUBNET-MASK>]} <ADDRESS-GROUP>	<p>Specifies the destination IPv4 address.</p> <ul style="list-style-type: none"> ■ any - specifies any destination IPv4 address. ■ <DST-IP-ADDRESS> - specifies the destination IPv4 host address. <ul style="list-style-type: none"> ○ <PREFIX-LENGTH> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 32. ○ <SUBNET-MASK> - specifies the address bits to mask (dotted decimal notation). ■ <ADDRESS-GROUP> - specifies an IPv4 address group that you defined earlier with <code>object-group ip address</code>.
[{eq gt lt} <PORT> range <MIN-PORT><MAX-PORT> group <PORT-GROUP>]	<p>Specifies the port, port range, or port group. Port numbers are in the range of 0 to 65535.</p> <ul style="list-style-type: none"> ■ eq <PORT> - specifies the Layer 4 port. ■ gt <PORT> - specifies any Layer 4 port greater than the indicated port. ■ lt <PORT> - specifies any Layer 4 port less than the indicated port. ■ range <MIN-PORT> <MAX-PORT> - specifies the Layer 4 port range. ■ group <PORT-GROUP> - specifies the Layer 4 port group that you defined earlier with <code>object-group port</code>. <p>NOTE: Upon application of the ACL, ACEs with L4 port ranges may consume more than one hardware entry.</p>

Parameter	Description
urg	Specifies matching on the TCP Flag: Urgent. (Applies only to the "in" (ingress) direction.)
ack	Specifies matching on the TCP Flag: Acknowledgment. (Applies only to the "in" (ingress) direction.)
psh	Specifies matching on the TCP Flag: Push buffered data to receiving application. (Applies only to the "in" (ingress) direction.)
rst	Specifies matching on the TCP Flag: Reset the connection. (Applies only to the "in" (ingress) direction.)
syn	Specifies matching on the TCP Flag: Synchronize sequence numbers. (Applies only to the "in" (ingress) direction.)
fin	Specifies matching on the TCP Flag: Finish connection. (Applies only to the "in" (ingress) direction.)
established	Specifies matching on the TCP Flag: Established connection. (Applies only to the "in" (ingress) direction.)
[icmp-type {echo echo-reply <ICMP-TYPE-VALUE>}]	Specifies the ICMP type. <ul style="list-style-type: none"> ■ echo - specifies an ICMP echo request packet. ■ echo-reply - specifies an ICMP echo reply packet. ■ <ICMP-TYPE-VALUE> - specifies an ICMP type value. Range: 0 to 255.
[icmp-code <ICMP-CODE-VALUE>]	Specifies the ICMP code value. Range: 0 to 255.
dscp <DSCP-SPECIFIER>	Specifies the Differentiated Services Code Point (DSCP), either a numeric <DSCP-VALUE> (0 to 63) or one of these keywords: <ul style="list-style-type: none"> ■ AF11 - DSCP 10 (Assured Forwarding Class 1, low drop probability) ■ AF12 - DSCP 12 (Assured Forwarding Class 1, medium drop probability) ■ AF13 - DSCP 14 (Assured Forwarding Class 1, high drop probability) ■ AF21 - DSCP 18 (Assured Forwarding Class 2, low drop probability) ■ AF22 - DSCP 20 (Assured Forwarding Class 2, medium drop probability) ■ AF23 - DSCP 22 (Assured Forwarding Class 2, high drop probability) ■ AF31 - DSCP 26 (Assured Forwarding Class 3, low drop probability) ■ AF32 - DSCP 28 (Assured Forwarding Class 3, medium drop probability) ■ AF33 - DSCP 30 (Assured Forwarding Class 3, high drop probability) ■ AF41 - DSCP 34 (Assured Forwarding Class 4, low drop probability) ■ AF42 - DSCP 36 (Assured Forwarding Class 4, medium drop probability) ■ AF43 - DSCP 38 (Assured Forwarding Class 4, high drop probability) ■ CS0 - DSCP 0 (Class Selector 0: Default) ■ CS1 - DSCP 8 (Class Selector 1: Scavenger) ■ CS2 - DSCP 16 (Class Selector 2: OAM) ■ CS3 - DSCP 24 (Class Selector 3: Signaling) ■ CS4 - DSCP 32 (Class Selector 4: Real time)

Parameter	Description
	<ul style="list-style-type: none"> ■ CS5 - DSCP 40 (Class Selector 5: Broadcast video) ■ CS6 - DSCP 48 (Class Selector 6: Network control) ■ CS7 - DSCP 56 (Class Selector 7) ■ EF - DSCP 46 (Expedited Forwarding)
ecn <ECN-VALUE>	Specifies an Explicit Congestion Notification value. Range: 0 to 3.
ip-precedence <IP-PRECEDENCE-VALUE>	Specifies an IP precedence value. Range: 0 to 7.
tos <TOS-VALUE>	Specifies the Type of Service value. Range: 0 to 31.
fragment	Specifies a fragment packet.
vlan <VLAN-ID>	Specifies VLAN tag to match on. 802.1Q VLAN ID. NOTE: This parameter cannot be used in any ACL that will be applied to a VLAN.
ttl <TTL-VALUE>	Specifies a time-to-live (hop limit) value. Range: 0 to 255.
count	Keeps the hit counts of the number of packets matching this ACE.
log	Keeps a log of the number of packets matching this ACE. Works with deny actions but not with permit actions. Works with ACLs applied on ingress or egress, except for control plane.
[<SEQUENCE-NUMBER>] comment <TEXT-STRING>	Adds a comment to an ACE. The no form removes only the comment from the ACE.

Usage

- If the <IP-PROTOCOL-NUM> parameter is used instead of a protocol name, ensure that any needed ACE-definition parameters specific to the selected protocol are also provided.
- When using multiple ACL types (IPv4, IPv6, or MAC) with logging on the same interface, the first packet that matches an ACE with log option is logged. Until the log-timer wait-period is over, any packets matching other ACL types do not create a log. At the end of the wait-period, the switch creates a summary log for all the ACLs that were matched, regardless of type.

Examples

Creating an IPv4 ACL with four entries:

```
switch(config)# access-list ip MY_IP_ACL
switch(config-acl-ip)# 10 permit udp any 172.16.1.0/24
switch(config-acl-ip)# 20 permit tcp 172.16.2.0/16 gt 1023 any
switch(config-acl-ip)# 30 permit tcp 172.26.1.0/24 any syn ack dscp 10
switch(config-acl-ip)# 40 deny any any any count
switch(config-acl-ip)# exit

switch(config)# show access-list
Type      Name
Sequence Comment
```



```

Action
Source IP Address
Destination IP Address
Additional Parameters
-----
IPv4      MY_IP_ACL
10 permit
   any
   172.16.1.0/255.255.255.0
20 permit
   172.16.2.0/255.255.0.0
   any
30 permit
   172.26.1.0/255.255.255.0
   any
   dscp: AF11
   ack
   syn
40 deny
   any
   any
Hit-counts: enabled
L3 Protocol
Source L4 Port(s)
Destination L4 Port(s)
-----
udp
tcp
> 1023
tcp
any

```

Adding a comment to an existing IPv4 ACE:

```

switch(config)# access-list ip MY_IP_ACL
switch(config-acl-ip)# 20 comment Permit all TCP ephemeral ports
switch(config-acl-ip)# exit

switch(config)# show access-list
Type      Name
Sequence  Comment
Action
Source IP Address
Destination IP Address
Additional Parameters
-----
IPv4      MY_IP_ACL
10 permit
   any
   172.16.1.0/255.255.255.0
20 Permit all TCP ephemeral ports
   permit
   172.16.2.0/255.255.0.0
   any
30 permit
   172.26.1.0/255.255.255.0
   any
   dscp: AF11
   ack
   syn
40 deny
   any
   any
Hit-counts: enabled
L3 Protocol
Source L4 Port(s)
Destination L4 Port(s)
-----
udp
tcp
> 1023
tcp
any

```

Removing a comment from an existing IPv4 ACE:

```

switch(config)# access-list ip MY_IP_ACL
switch(config-acl-ip)# no 20 comment
switch(config-acl-ip)# exit

switch(config)# show access-list
Type          Name
  Sequence Comment
      Action          L3 Protocol
      Source IP Address Source L4 Port(s)
      Destination IP Address Destination L4 Port(s)
      Additional Parameters
-----
IPv4          MY_IP_ACL
  10 permit          udp
      any
      172.16.1.0/255.255.255.0
  20 permit          tcp
      172.16.2.0/255.255.0.0
      any
      > 1023
  30 permit          tcp
      172.26.1.0/255.255.255.0
      any
      dscp: AF11
      ack
      syn
  40 deny            any
      any
      any
      Hit-counts: enabled

```

Adding an ACE (insert line 25) to an existing IPv4 ACL:

```

switch(config)# access-list ip MY_IP_ACL
switch(config-acl-ip)# 25 permit icmp 172.16.2.0/16 any
switch(config-acl-ip)# exit

switch(config)# show access-list
Type          Name
  Sequence Comment
      Action          L3 Protocol
      Source IP Address Source L4 Port(s)
      Destination IP Address Destination L4 Port(s)
      Additional Parameters
-----
IPv4          MY_IP_ACL
  10 permit          udp
      any
      172.16.1.0/255.255.255.0
  20 permit          tcp
      172.16.2.0/255.255.0.0
      any
      > 1023
  25 permit          icmp
      172.16.2.0/255.255.0.0 any
  30 permit          tcp
      172.26.1.0/255.255.255.0
      any
      dscp: AF11
      ack
      syn
  40 deny            any
      any
      any

```

```
any
any
Hit-counts: enabled
```

Replacing an ACE in an existing IPv4 ACL:

```
switch(config)# access-list ip MY_IP_ACL
switch(config-acl-ip)# 25 permit icmp 172.17.1.0/16 any
switch(config-acl-ip)# exit

switch(config)# show access-list
Type      Name
Sequence  Comment
          Action                L3 Protocol
          Source IP Address       Source L4 Port(s)
          Destination IP Address  Destination L4 Port(s)
          Additional Parameters
-----
IPv4      MY_IP_ACL
10 permit any                    udp
          172.16.1.0/255.255.255.0
20 permit 172.16.2.0/255.255.0.0 tcp
          any                    > 1023
25 permit 172.17.1.0/255.255.0.0 icmp
30 permit 172.26.1.0/255.255.255.0 tcp
          any
          dscp: AF11
          ack
          syn
40 deny  any                    any
          any
          any
          Hit-counts: enabled
```

Removing an ACE from an IPv4 ACL:

```
switch(config)# access-list ip MY_IP_ACL
switch(config-acl-ip)# no 25
switch(config-acl-ip)# exit

switch(config)# show access-list
Type      Name
Sequence  Comment
          Action                L3 Protocol
          Source IP Address       Source L4 Port(s)
          Destination IP Address  Destination L4 Port(s)
          Additional Parameters
-----
IPv4      MY_IP_ACL
10 permit any                    udp
          172.16.1.0/255.255.255.0
20 permit 172.16.2.0/255.255.0.0 tcp
          any                    > 1023
```

```

    any
30 permit
   172.26.1.0/255.255.255.0          tcp
    any
    dscp: AF11
    ack
    syn
40 deny                               any
    any
    any
Hit-counts: enabled

```

Copy an IPv4 ACL:

```

switch(config)# access-list ip MY_IP_ACL copy MY_IP_ACL2
switch(config)# show access-list
Type      Name
-----
Sequence  Comment
Action    L3 Protocol
Source IP Address  Source L4 Port(s)
Destination IP Address  Destination L4 Port(s)
Additional Parameters
-----
IPv4      MY_IP_ACL
10        permit          udp
         any
         172.16.1.0/255.255.255.0
20        permit          tcp
         172.16.2.0/255.255.0.0      > 1023
         any
30        permit          tcp
         172.26.1.0/255.255.255.0
         any
         dscp: AF11
         ack
         syn
40        deny          any
         any
         any
Hit-counts: enabled
-----
IPv4      MY_IP_ACL2
10        permit          udp
         any
         172.16.1.0/255.255.255.0
20        permit          tcp
         172.16.2.0/255.255.0.0      > 1023
         any
30        permit          tcp
         172.26.1.0/255.255.255.0
         any
         dscp: AF11
         ack

```

```

    syn
40 deny                    any
    any
    any
Hit-counts: enabled

```

Removing an IPv4 ACL:

```

switch(config)# no access-list ip MY_IP_ACL

switch(config)# show access-list
Type      Name
Sequence Comment
          Action                    L3 Protocol
          Source IP Address         Source L4 Port(s)
          Destination IP Address    Destination L4 Port(s)
          Additional Parameters
-----
IPv4      MY_IP_ACL2
1 permit                    udp
  any
  172.16.1.0/255.255.255.0
2 permit                    tcp
  172.16.2.0/255.255.0.0    > 1023
  any
3 permit                    tcp
  172.26.1.0/255.255.255.0
  any
  dscp: AF11
  ack
  syn
4 deny                    any
  any
  any
Hit-counts: enabled

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config The <code>access-list ip <ACL-NAME></code> command takes you into the named ACL context where you enter the ACEs.	Administrators or local user group members with execution rights for this command.

access-list ipv6

Syntax to create an IPv6 ACL and enter its context. Plus syntax to remove an ACL:

```
access-list ipv6 <ACL-NAME>
no access-list ipv6 <ACL-NAME>
```

Syntax (within the ACL context) for creating or removing ACEs for protocols ah, gre, esp, ospf, pim (ipv6 is available as an alias for any):

```
[<SEQUENCE-NUMBER>]
{permit|deny}
{any|ipv6|ah|gre|esp|ospf|pim|<IP-PROTOCOL-NUM>}
{any|<SRC-IP-ADDRESS>[/<PREFIX-LENGTH>]|<ADDRESS-GROUP>}
{any|<DST-IP-ADDRESS>[/<PREFIX-LENGTH>]|<ADDRESS-GROUP>}
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count] [log]

no <SEQUENCE-NUMBER>
```

Syntax (within the ACL context) for creating or removing ACEs for protocols sctp, tcp, udp:

```
[<SEQUENCE-NUMBER>]
{permit|deny}
{sctp|tcp|udp}
{any|<SRC-IP-ADDRESS>[/<PREFIX-LENGTH>]|<ADDRESS-GROUP>}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>|group <PORT-GROUP>}
{any|<DST-IP-ADDRESS>[/<PREFIX-LENGTH>]|<ADDRESS-GROUP>}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>|group <PORT-GROUP>}
[urg] [ack] [psh] [rst] [syn] [fin] [established]
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count] [log]

no <SEQUENCE-NUMBER>
```

Syntax (within the ACL context) for creating or removing ACEs for protocol icmpv6:

```
[<SEQUENCE-NUMBER>]
{permit|deny}
{icmpv6}
{any|<SRC-IP-ADDRESS>[/<PREFIX-LENGTH>]|<ADDRESS-GROUP>}
{any|<DST-IP-ADDRESS>[/<PREFIX-LENGTH>]|<ADDRESS-GROUP>}
[icmp-type {echo|echo-reply|<ICMP-TYPE-VALUE>}] [icmp-code <ICMP-CODE-VALUE>]
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count] [log]

no <SEQUENCE-NUMBER>
```

Syntax (within the ACL context) for ACE comments:

```
[<SEQUENCE-NUMBER>] comment <TEXT-STRING>

no <SEQUENCE-NUMBER> comment
```

Description

Creates an IPv6 Access Control List (ACL). The ACL is made of one or more Access Control Entries (ACEs) ordered and prioritized by sequence number. The lowest sequence number is the highest prioritized ACE. The `no` form of this command deletes the entire ACL, or deletes an ACE identified by sequence number, or deletes only the comment from the ACE identified by sequence number.

Parameter	Description
<ACL-NAME>	Specifies the name of this ACL.
<SEQUENCE-NUMBER>	Specifies a sequence number for the ACE. Range: 1 to 4294967295.
{permit deny}	Specifies whether to permit or deny traffic matching this ACE.

Parameter	Description
<code><IP-PROTOCOL-NUM></code>	Specifies the protocol as its Internet Protocol number. For example, 2 corresponds to the IGMP protocol. Range: 0 to 255.
<code>{any <SRC-IP-ADDRESS>[/<PREFIX-LENGTH>] <ADDRESS-GROUP>}</code>	Specifies the source IPv6 address. <ul style="list-style-type: none"> ■ any - specifies any source IPv6 address. ■ <SRC-IP-ADDRESS> - specifies the source IPv6 host address. <ul style="list-style-type: none"> ○ <PREFIX-LENGTH> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 128. ■ <ADDRESS-GROUP> - specifies an IPv6 address group that you defined earlier with <code>object-group ipv6 address</code>.
<code>{any <DST-IP-ADDRESS>[/<PREFIX-LENGTH>] <ADDRESS-GROUP>}</code>	Specifies the destination IPv6 address. <ul style="list-style-type: none"> ■ any - specifies any destination IPv6 address. ■ <DST-IP-ADDRESS> - specifies the destination IPv6 host address. <ul style="list-style-type: none"> ○ <PREFIX-LENGTH> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 128. ■ <ADDRESS-GROUP> - specifies an IPv6 address group that you defined earlier with <code>object-group ipv6 address</code>.
<code>[[eq gt lt] <PORT> range <MIN-PORT><MAX-PORT> group <PORT-GROUP>]</code>	Specifies the port, port range, or port group. Port numbers are in the range of 0 to 65535. <ul style="list-style-type: none"> ■ eq <PORT> - specifies the Layer 4 port. ■ gt <PORT> - specifies any Layer 4 port greater than the indicated port. ■ lt <PORT> - specifies any Layer 4 port less than the indicated port. ■ range <MIN-PORT> <MAX-PORT> - specifies the Layer 4 port range. ■ group <PORT-GROUP> - specifies the Layer 4 port group that you defined earlier with <code>object-group port</code>. <p>NOTE: Upon application of the ACL, ACEs with L4 port ranges may consume more than one hardware entry.</p>
<code>urg, ack, psh, rst, syn, fin, established</code>	These TCP flag-matching parameters are not supported.
<code>[icmp-type {echo echo-reply <ICMP-TYPE-VALUE>}]</code>	Specifies the ICMP type. <ul style="list-style-type: none"> ■ echo - specifies an ICMP echo request packet. ■ echo-reply - specifies an ICMP echo reply packet. ■ <ICMP-TYPE-VALUE> - specifies an ICMP type value. Range: 0 to 255.
<code>[icmp-code <ICMP-CODE-VALUE>]</code>	Specifies the ICMP code value. Range: 0 to 255.

Parameter	Description
dscp <DSCP-SPECIFIER>	<p>Specifies the Differentiated Services Code Point (DSCP), either a numeric <DSCP-VALUE> (0 to 63) or one of these keywords:</p> <ul style="list-style-type: none"> ■ AF11 - DSCP 10 (Assured Forwarding Class 1, low drop probability) ■ AF12 - DSCP 12 (Assured Forwarding Class 1, medium drop probability) ■ AF13 - DSCP 14 (Assured Forwarding Class 1, high drop probability) ■ AF21 - DSCP 18 (Assured Forwarding Class 2, low drop probability) ■ AF22 - DSCP 20 (Assured Forwarding Class 2, medium drop probability) ■ AF23 - DSCP 22 (Assured Forwarding Class 2, high drop probability) ■ AF31 - DSCP 26 (Assured Forwarding Class 3, low drop probability) ■ AF32 - DSCP 28 (Assured Forwarding Class 3, medium drop probability) ■ AF33 - DSCP 30 (Assured Forwarding Class 3, high drop probability) ■ AF41 - DSCP 34 (Assured Forwarding Class 4, low drop probability) ■ AF42 - DSCP 36 (Assured Forwarding Class 4, medium drop probability) ■ AF43 - DSCP 38 (Assured Forwarding Class 4, high drop probability) ■ CS0 - DSCP 0 (Class Selector 0: Default) ■ CS1 - DSCP 8 (Class Selector 1: Scavenger) ■ CS2 - DSCP 16 (Class Selector 2: OAM) ■ CS3 - DSCP 24 (Class Selector 3: Signaling) ■ CS4 - DSCP 32 (Class Selector 4: Real time) ■ CS5 - DSCP 40 (Class Selector 5: Broadcast video) ■ CS6 - DSCP 48 (Class Selector 6: Network control) ■ CS7 - DSCP 56 (Class Selector 7) ■ EF - DSCP 46 (Expedited Forwarding)
ecn <ECN-VALUE>	Specifies an Explicit Congestion Notification value. Range: 0-3.
ip-precedence <IP-PRECEDENCE-VALUE>	Specifies an IP precedence value. Range: 0-7.
tos <TOS-VALUE>	Specifies the Type of Service value. Range: 0-31.
fragment	Not supported.
vlan <VLAN-ID>	Not supported.
ttl <TTL-VALUE>	Not supported.

Parameter	Description
count	Keeps the hit counts of the number of packets matching this ACE.
log	Keeps a log of the number of packets matching this ACE. Works with <code>deny</code> actions but not with <code>permit</code> actions. Works with ACLs applied on ingress or egress, except for control plane.
[<SEQUENCE-NUMBER>] comment <TEXT-STRING>	Adds a comment to an ACE. The <code>no</code> form removes only the comment from the ACE.

Usage

- If the <IP-PROTOCOL-NUM> parameter is used instead of a protocol name, ensure that any needed ACE-definition parameters specific to the selected protocol are also provided.



The 8400 switch cannot match on MLD (Multicast Listener Discovery) packets using the ICMPv6 protocol number 58 due to the Hop-by-Hop options extension header which is present on MLD packets. The presence of the extension header in the packet causes the 8400 to match on protocol number 0 (HOPOPT) and not protocol number 58 (ICMPv6). Therefore, to achieve the wanted matchings, specify protocol 0 (NOT 58) in your ACEs.

For example, to permit the joins from `ff12::01` using protocol 0 in the ACE:

```
10 permit 0 any ff12::0/64
```

- When using multiple ACL types (IPv4, IPv6, or MAC) with logging on the same interface, the first packet that matches an ACE with `log` option is logged. Until the log-timer wait-period is over, any packets matching other ACL types do not create a log. At the end of the wait-period, the switch creates a summary log all the ACLs that were matched, regardless of type.

Examples

Creating an IPv6 ACL with four entries:

```
switch(config)# access-list ipv6 MY_IPV6_ACL
switch(config-acl-ipv6)# 10 permit udp any 2001::1/64
switch(config-acl-ipv6)# 20 permit tcp 2001:2001::2:1/128 gt 1023 any
switch(config-acl-ipv6)# 30 permit tcp 2001:2011::1/64 any
switch(config-acl-ipv6)# 40 deny any any any count
switch(config-acl-ipv6)# exit

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action
          Source IP Address
          Destination IP Address
          Additional Parameters
-----
IPv6      MY_IPV6_ACL
10 permit                                udp
   any
   2001::1/64
20 permit                                tcp
   2001:2001::2:1
   > 1023
```

```

    any
30 permit                               tcp
   2001:2011::1/64
    any
40 deny                                 any
   any
   any
Hit-counts: enabled

```

Adding a comment to an existing IPv6 ACE:

```

switch(config)# access-list ipv6 MY_IPV6_ACL
switch(config-acl-ipv6)# 20 comment Permit all TCP ephemeral ports
switch(config-acl-ipv6)# exit

```

```

switch(config)# do show access-list

```

Type	Name	Sequence	Action	L3 Protocol
			Source IP Address	Source L4 Port(s)
			Destination IP Address	Destination L4 Port(s)
			Additional Parameters	

IPv6	MY_IPV6_ACL			
		10	permit any 2001::1/64	udp
		20	Permit all TCP ephemeral ports permit 2001:2001::2:1 any	tcp > 1023
		30	permit 2001:2011::1/64 any	tcp
		40	deny any any	any
			Hit-counts: enabled	

Removing a comment from an existing IPv6 ACE:

```

switch(config)# access-list ipv6 MY_IPV6_ACL
switch(config-acl-ipv6)# no 20 comment
switch(config-acl-ipv6)# exit

```

```

switch(config)# do show access-list

```

Type	Name	Sequence	Action	L3 Protocol
			Source IP Address	Source L4 Port(s)
			Destination IP Address	Destination L4 Port(s)
			Additional Parameters	

IPv6	MY_IPV6_ACL			
		10	permit any 2001::1/64	udp
		20	permit 2001:2001::2:1	tcp > 1023

```

    any
30 permit                                tcp
    2001:2011::1/64
    any
40 deny                                  any
    any
    any
Hit-counts: enabled

```

Adding an ACE to an existing IPv6 ACL:

```

switch(config)# access-list ipv6 MY_IPV6_ACL
switch(config-acl-ipv6)# 25 permit icmpv6 2001::1/64 any
switch(config-acl-ipv6)# exit

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action                L3 Protocol
          Source IP Address      Source L4 Port(s)
          Destination IP Address  Destination L4 Port(s)
          Additional Parameters
-----
IPv6      MY_IPV6_ACL
10 permit                                udp
    any
    2001::1/64
20 permit                                tcp
    2001:2001::2:1                    > 1023
    any
25 permit                                icmpv6
    2001::1/64
    any
30 permit                                tcp
    2001:2011::1/64
    any
40 deny                                  any
    any
    any
Hit-counts: enabled

```

Replacing an ACE in an existing IPv6 ACL:

```

switch(config)# access-list ipv6 MY_IPV6_ACL
switch(config-acl-ipv6)# 25 permit icmpv6 2001::2:1/64 any
switch(config-acl-ipv6)# exit

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action                L3 Protocol
          Source IP Address      Source L4 Port(s)
          Destination IP Address  Destination L4 Port(s)
          Additional Parameters
-----
IPv6      MY_IPV6_ACL
10 permit                                udp
    any
    2001::1/64

```

```

20 permit                tcp
   2001:2001::2:1        > 1023
   any
25 permit                icmpv6
   2001::2:1/64
   any
30 permit                tcp
   2001:2011::1/64
   any
40 deny                  any
   any
   any
Hit-counts: enabled

```

Removing an ACE from an IPv6 ACL:

```

switch(config)# access-list ipv6 MY_IPV6_ACL
switch(config-acl-ipv6)# no 25
switch(config-acl-ipv6)# exit

switch(config)# do show access-list
Type          Name
Sequence      Comment
Action                               L3 Protocol
Source IP Address                     Source L4 Port(s)
Destination IP Address                 Destination L4 Port(s)
Additional Parameters
-----
IPv6          MY_IPV6_ACL
10 permit                udp
   any
   2001::1/64
20 permit                tcp
   2001:2001::2:1        > 1023
   any
30 permit                tcp
   2001:2011::1/64
   any
40 deny                  any
   any
   any
Hit-counts: enabled

```

Removing an IPv6 ACL:

```

switch(config)# no access-list ipv6 MY_IPV6_ACL

switch(config)# do show access-list
Type          Name
Sequence      Comment
Action                               L3 Protocol
Source IP Address                     Source L4 Port(s)
Destination IP Address                 Destination L4 Port(s)
Additional Parameters
-----
IPv6          MY_IPV6_ACL2
1 permit                udp
   any
   2001::1/64

```

```

2 Permit all TCP ephemeral ports
  permit                tcp
  2001:2001::2:1        > 1023
  any
3 permit                tcp
  2001:2011::1/64
  any
4 deny                  any
  any
  any
Hit-counts: enabled

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config The <code>access-list ipv6 <ACL-NAME></code> command takes you into the named ACL context where you enter the ACEs.	Administrators or local user group members with execution rights for this command.

access-list mac

```

access-list mac <ACL-NAME>
no access-list mac <ACL-NAME>

```

```

[<SEQUENCE-NUMBER>]
{permit|deny}
{any|<SRC-MAC-ADDRESS>[/<ETHERNET-MASK>]}
{any|<DST-MAC-ADDRESS>[/<ETHERNET-MASK>]}
{any|arp|appletalk|arp|fcoe|fcoe-init|ip|ipv6|
  ipx-arpa|ipx-non-arpa|is-is|lldp|mpls-multicast|mpls-unicast|q-in-q|
  rbridge|trill|wake-on-lan|<NUMERIC-ETHERTYPE>}
[pcp <PCP-VALUE>] [vlan <VLAN-ID>] [count] [log]
no <SEQUENCE-NUMBER>

```

```

[<SEQUENCE-NUMBER>] comment <TEXT-STRING>
no <SEQUENCE-NUMBER> comment

```

Description

Creates a MAC Access Control List (ACL). The ACL is made of one or more Access Control Entries (ACEs) ordered and prioritized by sequence numbers. The lowest sequence number is the highest prioritized ACE. The `no` form of this command deletes the entire ACL, or deletes an ACE identified by sequence number, or deletes only the comment from the ACE identified by sequence number.

Parameter	Description
<ACL-NAME>	Specifies the name of this ACL.
<SEQUENCE-NUMBER>	Specifies a sequence number for the ACE. Range: 1 to 4294967295.
{permit deny}	Specifies whether to permit or deny traffic matching this ACE.
comment	Specifies storing the remaining entered text as an ACE comment.
{any <SRC-MAC-ADDRESS>[/<ETHERNET-MASK>]}	Specifies the source host MAC address (xxxx.xxxx.xxxx), OUI, or the keyword any. You can optionally include the following: <ETHERNET-MASK> - The address bits to mask (xxxx.xxxx.xxxx).
{any <DST-MAC-ADDRESS>[/<ETHERNET-MASK>]}	Specifies the destination host MAC address (xxxx.xxxx.xxxx), OUI, or the keyword any. You can optionally include the following: <ETHERNET-MASK> - The address bits to mask (xxxx.xxxx.xxxx).
{any arp appletalk ... wake-on-lan <NUMERIC-ETHERTYPE>	Specifies the protocol encapsulated in the Ethernet frame. The encapsulated protocol is identified by the EtherType Ethernet field. The EtherType is specified in one of the following three ways: <ul style="list-style-type: none"> ■ any - any EtherType. ■ <NUMERIC-ETHERTYPE> - the numerical EtherType protocol number. Range: 0x600 to 0xffff. ■ One of these EtherType protocol name keywords: <ul style="list-style-type: none"> ○ arp ○ appletalk ○ arp ○ fcoe ○ fcoe-init ○ ip ○ ipv6 ○ ipx-arpa ○ ipx-non-arpa ○ is-is ○ lldp ○ mpls-multicast ○ mpls-unicast ○ q-in-q ○ rbridge ○ trill ○ wake-on-lan
pcp <PCP-VALUE>	Specifies 802.1Q QoS Priority Code Point value. Range: 0 to 7.
vlan <VID>	Specifies a VLAN ID. The VLAN ID must exist.

Parameter	Description
	NOTE: This parameter cannot be used in any ACL that will be applied to a VLAN.
count	Keeps the hit counts of the number of packets matching this ACE.
log	Keeps a log of the number of packets matching this ACE. Works with <code>deny</code> actions but not with <code>permit</code> actions. Works with ACLs applied on ingress or egress.

Usage

When using multiple ACL types (IPv4, IPv6, or MAC) with logging on the same interface, the first packet that matches an ACE with `log` option is logged. Until the log-timer wait-period is over, any packets matching other ACL types do not create a log. At the end of the wait-period, the switch creates a summary log all the ACLs that were matched, regardless of type.

Examples

Creating a MAC ACL with four entries:

```
switch(config)# access-list mac MY_MAC_ACL
switch(config-acl-ip)# 10 permit 1122.3344.5566/ffff.ffff.0000 any ipv6
switch(config-acl-ip)# 20 permit aaaa.bbbb.cccc 1111.2222.3333 any pcp 4
switch(config-acl-ip)# 30 permit any any appletalk vlan 40
switch(config-acl-ip)# 40 deny any any any count
switch(config-acl-ip)# exit

switch(config)# do show access-list
Type      Name
Sequence Comment
          Action                      EtherType
          Source MAC Address
          Destination MAC Address
          Additional Parameters
-----
MAC       MY_MAC_ACL
10 permit 1122.3344.5566/ffff.ffff.0000  ipv6
   any
20 permit aaaa.bbbb.cccc 1111.2222.3333  any
   QoS Priority Code Point: 4
30 permit any any appletalk               appletalk
   any
   VLAN: 40
40 deny  any any any                       any
   any
   Hit-counts: enabled
```

Adding a comment to an existing MAC ACE:

```

switch(config)# access-list mac MY_MAC_ACL
switch(config-acl-ip)# 30 comment Permit all vlan-40 tagged Appletalk traffic
switch(config-acl-ip)# exit

switch(config)# do show access-list
Type          Name
  Sequence Comment
            Action                      EtherType
            Source MAC Address
            Destination MAC Address
            Additional Parameters
-----
MAC          MY_MAC_ACL
  10 permit
    1122.3344.5566/ffff.ffff.0000
    any
  20 permit
    aaaa.bbbb.cccc
    1111.2222.3333
    QoS Priority Code Point: 4
  30 Permit all vlan-40 tagged Appletalk traffic
    permit
    any
    any
    VLAN: 40
  40 deny
    any
    any
    Hit-counts: enabled

```

Removing a comment from an existing MAC ACE:

```

switch(config)# access-list mac MY_MAC_ACL
switch(config-acl-mac)# no 30 comment
switch(config-acl-mac)# exit

switch(config)# do show access-list
Type          Name
  Sequence Comment
            Action                      EtherType
            Source MAC Address
            Destination MAC Address
            Additional Parameters
-----
MAC          MY_MAC_ACL
  10 permit
    1122.3344.5566/ffff.ffff.0000
    any
  20 permit
    aaaa.bbbb.cccc
    1111.2222.3333
    QoS Priority Code Point: 4
  30 permit
    any
    any
    VLAN: 1
  40 deny
    any
    any
    Hit-counts: enabled

```

Adding an ACE to an existing MAC ACL:


```

switch(config)# access-list mac MY_MAC_ACL
switch(config-acl-ip)# 35 permit any aabb.cc11.1234 0xffee
switch(config-acl-ip)# exit

```

```

switch(config)# do show access-list

```

```

Type      Name
  Sequence Comment
           Action                EtherType
           Source MAC Address
           Destination MAC Address
           Additional Parameters
-----
MAC       MY_MAC_ACL
  10 permit                ipv6
           1122.3344.5566/ffff.ffff.0000
           any
  20 permit                any
           aaaa.bbbb.cccc
           1111.2222.3333
           QoS Priority Code Point: 4
  30 permit                appletalk
           any
           any
           VLAN: 1
  35 permit                0xffee
           any
           aabb.cc11.1234
  40 deny                  any
           any
           Hit-counts: enabled

```

Replacing an ACE in an existing MAC ACL:

```

switch(config)# access-list mac MY_MAC_ACL
switch(config-acl-ip)# 35 permit any aabb.cc11.1234 0xeeee
switch(config-acl-ip)# exit

```

```

switch(config)# do show access-list

```

```

Type      Name
  Sequence Comment
           Action                EtherType
           Source MAC Address
           Destination MAC Address
           Additional Parameters
-----
MAC       MY_MAC_ACL
  10 permit                ipv6
           1122.3344.5566/ffff.ffff.0000
           any
  20 permit                any
           aaaa.bbbb.cccc
           1111.2222.3333
           QoS Priority Code Point: 4
  30 permit                appletalk
           any
           any
           VLAN: 1
  35 permit                0xeeee
           any
           aabb.cc11.1234

```

```

40 deny
   any
   any
Hit-counts: enabled

```

Removing an ACE from an MAC ACL:

```

switch(config)# access-list mac MY_MAC_ACL
switch(config-acl-ip)# no 35
switch(config-acl-ip)# exit

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action                      EtherType
          Source MAC Address
          Destination MAC Address
          Additional Parameters
-----
MAC       MY_MAC_ACL
10 permit
   1122.3344.5566/ffff.ffff.0000
   any                               ipv6
20 permit
   aaaa.bbbb.cccc
   1111.2222.3333
   QoS Priority Code Point: 4
30 permit
   any
   any
   VLAN: 1
   any                               appletalk
40 deny
   any
   any
Hit-counts: enabled

```

Removing a MAC ACL:

```

switch(config)# no access-list mac MY_MAC_ACL

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action                      EtherType
          Source MAC Address
          Destination MAC Address
          Additional Parameters
-----
MAC       MY_MAC_ACL2
1 permit
   1122.3344.5566/ffff.ffff.0000
   any                               ipv6
2 permit
   aaaa.bbbb.cccc
   1111.2222.3333
   QoS Priority Code Point: 4
3 Permit all vlan-40 tagged Appletalk traffic
   permit
   any                               appletalk

```

```

any
VLAN: 1
4 deny any any
Hit-counts: enabled

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config The <code>access-list mac <ACL-NAME></code> command takes you into the named ACL context where you enter the ACEs.	Administrators or local user group members with execution rights for this command.

access-list resequence

```
access-list {ip|ipv6|mac} <ACL-NAME> resequence <STARTING-SEQUENCE-NUMBER> <INCREMENT>
```

Description

Resequences the ACE sequence numbers in an ACL.

Parameter	Description
{ip ipv6 mac}	Specifies the ACL type.
<ACL-NAME>	Specifies the ACL name.
<STARTING-SEQUENCE-NUMBER>	Specifies the starting sequence number.
<INCREMENT>	Specifies the sequence number increment.

Examples

Resequencing an IPv4 ACL to start at 1 with an increment of 1:

```

switch(config)# access-list ip MY_IP_ACL resequence 1 1
switch(config-acl-ip)# exit

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action          L3 Protocol
          Source IP Address  Source L4 Port(s)
          Destination IP Address  Destination L4 Port(s)

```

```

Additional Parameters
-----
IPv4      MY_IP_ACL
1 permit          udp
   any
   172.16.1.0/255.255.255.0
2 permit          tcp
   172.16.2.0/255.255.0.0
   any            > 1023
3 permit          tcp
   172.26.1.0/255.255.255.0
   any
   dscp: AF11
   ack
   syn
4 deny           any
   any
   any
Hit-counts: enabled

```

Resequencing an IPv6 ACL to start at 1 with an increment of 1:

```

switch(config)# access-list ipv6 MY_IPV6_ACL resequence 1 1
switch(config-acl-ip)# exit

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action          L3 Protocol
          Source IP Address Source L4 Port(s)
          Destination IP Address Destination L4 Port(s)
          Additional Parameters
-----
IPv6      MY_IPV6_ACL
1 permit          udp
   any
   2001::1/64
2 Permit all TCP ephemeral ports
   permit          tcp
   2001:2001::2:1  > 1023
   any
3 permit          tcp
   2001:2011::1/64
   any
4 deny           any
   any
   any
Hit-counts: enabled

```

Resequencing a MAC ACL to start at 1 with an increment of 1:

```

switch(config)# access-list mac MY_MAC_ACL resequence 1 1
switch(config-acl-mac)# exit

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action          EtherType
          Source MAC Address

```

```

Destination MAC Address
Additional Parameters
-----
MAC      MY_MAC_ACL
1 permit          ipv6
  1122.3344.5566/ffff.ffff.0000
  any
2 permit          any
  aaaa.bbbb.cccc
  1111.2222.3333
  QoS Priority Code Point: 4
3 Permit all vlan-40 tagged Appletalk traffic
  permit          appletalk
  any
  any
  VLAN: 1
4 deny           any
  any
  any
Hit-counts: enabled

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

access-list reset

```
access-list {all|ip <ACL-NAME>|ipv6 <ACL-NAME>|mac <ACL-NAME>} reset
```

Description

Changes the user-specified ACL configuration to match the active ACL configuration. Use this command when a discrepancy exists between what the user configured and what is active and accepted by the system.

Parameter	Description
all ip <ACL-NAME> ipv6 <ACL-NAME> mac <ACL-NAME>	Specifies one of the following: <ul style="list-style-type: none"> ■ a reset of all ACLs. ■ a reset of a named IPv4 ACL. ■ a reset of a named IPv6 ACL. ■ a reset of a named MAC ACL.

Usage

The output of the `show access-list` command displays the active configuration of the product. The active configuration is the ACLs that have been configured and accepted by the system. The output of the `show`

`access-list` command with the `configuration` parameter, displays the ACLs that have been configured. The output of this command may not be the same as what was programmed in hardware or what is active on the product.

If the active ACLs and user-configured ACLs are not the same, a warning message is displayed in the output of the `show` command. Modify the user-configured ACL until the warning message is no longer displayed or run the `access-list reset` command to change the user-specified configuration to match the active configuration.

Examples

Apply an ACL with TCP acknowledgments (ACKs) on ingress, which is unsupported by hardware:

```
switch(config-acl)# 10 permit tcp 172.16.2.0/16 any ack
```

Displaying the user-specified configuration:

```
switch(config)# do show access-list commands
! access-list ip TEST_ACL user configuration does not match active configuration.
! run 'access-list TYPE NAME reset' to reset access-list to match active
configuration.
access-list ip TEST_ACL
! access-list ip TEST_ACL user configuration does not match active configuration.
! run 'access-list TYPE NAME reset' to reset access-list to match active
configuration.
interface 1/1/1
    apply access-list ip TEST_ACL in

switch(config)# do show access-list commands configuration
! access-list ip TEST_ACL user configuration does not match active configuration.
! run 'access-list TYPE NAME reset' to reset access-list to match active
configuration.
access-list ip TEST_ACL
    10 permit tcp 172.16.2.0/255.255.0.0 any ack
! access-list ip TEST_ACL user configuration does not match active configuration.
! run 'access-list TYPE NAME reset' to reset access-list to match active
configuration.
interface 1/1/1
    apply access-list ip TEST_ACL in

switch(config)# do show access-list
Type          Name
Sequence      Comment
              Action                    L3 Protocol
              Source IP Address              Source L4 Port(s)
              Destination IP Address      Destination L4 Port(s)
              Additional Parameters
-----
% Warning: TEST_ACL user configuration does not match active configuration.
%      run 'access-list TYPE NAME reset' to reset access-list to match active
configuration.
IPv4          TEST_ACL

switch(config)# do show access-list configuration
Type          Name
Sequence      Comment
              Action                    L3 Protocol
              Source IP Address              Source L4 Port(s)
              Destination IP Address      Destination L4 Port(s)
              Additional Parameters
```

```

-----
% Warning: TEST_ACL user configuration does not match active configuration.
%      run 'access-list TYPE NAME reset' to reset access-list to match active
configuration.
IPv4      TEST_ACL
          10
            permit          tcp
            172.16.2.0/255.255.0.0
            any
            ack

```

Resetting the user-specified configuration to match the active configuration.

```
switch(config)# access-list ip TEST_ACL reset
```

Displaying the updated user-specified configuration.

```

switch(config)# do show access-list commands
access-list ip TEST_ACL
interface 1/1/1
  apply access-list ip TEST_ACL in

switch(config)# do show access-list commands configuration
access-list ip TEST_ACL
interface 1/1/1
  apply access-list ip TEST_ACL in

switch(config)# do show access-list
Type      Name
Sequence  Comment
          Action          L3 Protocol
          Source IP Address Source L4 Port(s)
          Destination IP Address Destination L4 Port(s)
          Additional Parameters
-----
IPv4      TEST_ACL

switch(config)# do show access-list configuration
Type      Name
Sequence  Comment
          Action          L3 Protocol
          Source IP Address Source L4 Port(s)
          Destination IP Address Destination L4 Port(s)
          Additional Parameters
-----
IPv4      TEST_ACL

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

apply access-list control-plane

```
apply access-list {ip|ipv6} <ACL-NAME> control-plane vrf <VRF-NAME>
no apply access-list {ip|ipv6} <ACL-NAME> control-plane vrf <VRF-NAME>
```

Description

Applies an ACL to the specified VRF.

The `no` form of this command removes application of the ACL from the specified VRF.

Parameter	Description
<code>ip ipv6</code>	Specifies the ACL type: <code>ip</code> for IPv4, or <code>ipv6</code> for IPv6.
<code><ACL-NAME></code>	Specifies the ACL name.
<code>vrf <VRF-NAME></code>	Specifies the VRF name.

Usage

Only one ACL per type (`ip`, or `ipv6`) may be applied to a control plane VRF at a time. Therefore, using the `apply access-list control-plane` command on a VRF with an already-applied ACL of the same type, will replace the applied ACL.

Examples

Applying `My_ip_ACL` to control plane traffic on the default VRF:

```
switch(config)# apply access-list ip My_ip_ACL control-plane vrf default
```

Replacing `My_ip_ACL` with `My_Replacement_ACL` on the default VRF:

```
switch(config)# apply access-list ip My_Replacement_ACL control-plane vrf default
```

Remove (unapply) the `My_Replacement_ACL` from the default VRF. Any other interfaces or VLANs with `My_Replacement_ACL` applied are unaffected.

```
switch(config)# no apply access-list ip My_Replacement_ACL control-plane vrf default
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

apply access-list (to interface or LAG)

```
apply access-list {ip | ipv6 | mac} <ACL-NAME> {in | out}
no apply access-list {ip | ipv6 | mac} <ACL-NAME> {in | out}
```

Description

Applies an ACL to the interface (Individual front plane port) or Link Aggregation Group (LAG) identified by the current interface or LAG context.

The `no` form of this command removes application of the ACL from the current interface or LAG identified by the current interface or LAG context.

Parameter	Description
<code>ip ipv6 mac</code>	Specifies the ACL type: <code>ip</code> for IPv4, <code>ipv6</code> for IPv6, or <code>mac</code> for MAC ACL.
<code><ACL-NAME></code>	Specifies the ACL name.
<code>in</code>	Selects the inbound (ingress) traffic direction.
<code>out</code>	Selects the outbound (egress) traffic direction. Only for IPv4 ACLs applied to route-only ports. Not available for ACLs applied to IPv4 bridged ports, IPv6 ports, or MAC ACLs applied to ports.

Usage

- Each ACL of a given type can be applied to the same interface or LAG once in each direction. Therefore, using the `apply access-list` command on an interface or LAG with an already-applied ACL of the same type and direction will replace the currently applied ACL.
- An ACL can be applied to an individual front plane port or to a Link Aggregation Group (LAG).
- A port that is a member of a LAG with an applied ACL cannot have a different ACL applied to that member port.
- When the port membership of a LAG with an applied ACL is changed, the LAG ACL is automatically applied or removed from that port depending on the modification type.
- No ACLs (including ACLs for IPv4, IPv6, and MAC) are supported in egress on the Layer 2 interface. Egress ACLs can only be applied to Layer 3 (route-only) interfaces. Applying an egress ACL to a Layer 2 interface results in an error.

Examples

Applying `My_IP_ACL` to ingress traffic on interface range `1/1/10` to `1/1/12`:

```
switch(config)# int 1/1/10-1/1/12
switch((config-if-<1/1/10-1/1/12>)# apply access-list ip My_IP_ACL in
switch((config-if-<1/1/10-1/1/12>)# exit
```

Applying MY_IP_ACL to ingress traffic on LAG 100 and egress traffic on interface 1/1/2:

```
switch(config)# interface lag 100
switch(config-lag-if)# apply access-list ip MY_IP_ACL in
switch(config-lag-if)# exit

switch(config)# interface 1/1/2
switch(config-if)# apply access-list ip MY_IP_ACL out
switch(config-if)# exit
switch(config)#
```

Applying MY_IPV6_ACL to ingress traffic on interface 1/1/1 and to ingress traffic on LAG 100:

```
switch(config)# interface 1/1/1
switch(config-if)# apply access-list ipv6 MY_IPV6_ACL in
switch(config-if)# exit

switch(config)# interface lag 100
switch(config-lag-if)# apply access-list ipv6 MY_IPV6_ACL in
switch(config-lag-if)# exit
switch(config)#
```

Applying MY_MAC_ACL to ingress traffic on interface 1/1/1 and ingress traffic on interface 1/1/2:

```
switch(config)# interface 1/1/1
switch(config-if)# apply access-list mac MY_MAC_ACL in
switch(config-if)# exit

switch(config)# interface 1/1/2
switch(config-if)# apply access-list mac MY_MAC_ACL in
switch(config-if)# exit
switch(config)#
```

Replacing MY_IP_ACL with MY_REPLACEMENT_ACL on interface 1/1/2:

```
switch(config)# interface 1/1/2
switch(config-if)# apply access-list ip MY_REPLACEMENT_ACL out
switch(config-if)# exit
switch(config)#
```

Unapplying MY_REPLACEMENT_ACL from interface 1/1/2 (out):

```
switch(config)# interface 1/1/2
switch(config-if)# no apply access-list ip MY_REPLACEMENT_ACL out
switch(config-if)# exit
switch(config)#
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if config-lag-if	Administrators or local user group members with execution rights for this command.

apply access-list (to interface VLAN)

```
apply access-list {ip|ipv6} <ACL-NAME> routed-in
no apply access-list {ip|ipv6} <ACL-NAME> routed-in
```

Description

Applies an ACL to the interface VLAN (or range of interface VLANs) identified by the current interface VLAN context. Using the `apply access-list` command on an interface VLAN interface with an already-applied ACL of the same type will replace the currently-applied ACL.

The `no` form of this command removes application of the ACL from the interface VLAN (or range of interface VLANs) identified by the current interface VLAN context.

Parameter	Description
ip ipv6	Specifies the ACL type: <code>ip</code> for IPv4, <code>ipv6</code> for IPv6.
<ACL-NAME>	Specifies the ACL name.
routed-in	Selects the routed inbound (routed ingress) traffic direction.

Usage

- Each ACL of a given type can be applied to the same interface VLAN once. Therefore, using the `apply access-list` command on an interface VLAN with an already-applied ACL of the same type, will replace the applied ACL.
- When an ACL is applied to an interface VLAN, it will create hardware entries on all line cards regardless of whether an interface VLAN member exists on any specific line card.

Examples

Creating an IPv4 ACL and applying it to routed ingress traffic on interface VLAN `vlan100`:

```
switch(config)# access-list ip test
switch(config-acl-ip)# 10 permit any 1.1.1.2 2.2.2.2 count
switch(config-acl-ip)# 20 permit any 1.1.1.2 2.2.2.1 count
switch(config-acl-ip)# 30 permit any 2.2.2.2 1.1.1.2 count
switch(config-acl-ip)# 40 permit any 2.2.2.2 1.1.1.1 count
switch(config-acl-ip)# 50 permit any any any count
switch(config-acl-ip)# exit
switch(config)#
switch(config)# interface vlan100
switch(config-if-vlan)# apply access-list ip test routed-in
```

Applying `My_ip_ACL` to routed ingress traffic on interface VLAN 10:

```
switch(config)# interface vlan 10
switch(config-if-vlan)# apply access-list ip My_ip_ACL routed-in
```

Applying `My_ipv6_ACL` to routed ingress traffic on interface VLAN 10:

```
switch(config)# interface vlan 10
switch(config-if-vlan)# apply access-list ipv6 My_ip_ACL routed-in
```

Applying My_ip_ACL to routed ingress traffic on interface VLANs 20 to 25:

```
switch(config)# interface vlan 20-25
switch(config-if-vlan-<20-25>)# apply access-list ip My_ip_ACL routed-in
```

Replacing My_ipv6_ACL with My_Replacement_ACL on interface VLAN 10 (following the above examples):

```
switch(config)# interface vlan 10
switch(config-if-vlan)# apply access-list ipv6 My_Replacement_ACL routed-in
```

Removing (unapplying) My_Replacement_ACL on interface VLAN 10. Any other interfaces or VLANs with My_Replacement_ACL applied are not affected:

```
switch(config)# interface vlan 10
switch(config-if-vlan)# no apply access-list ipv6 My_Replacement_ACL routed-in
```

Removing (unapplying) My_ip_ACL on interface VLANs 20 to 25. Any other interfaces or VLANs with My_ip_ACL applied are not affected:

```
switch(config)# interface vlan 20-25
switch(config-if-vlan-<20-25>)# no apply access-list ip My_ip_ACL routed-in
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
8400	config-if-vlan	Administrators or local user group members with execution rights for this command.

apply access-list (to VLAN)

```
apply access-list {ip|ipv6|mac} <ACL-NAME> in
no apply access-list {ip|ipv6|mac} <ACL-NAME> in
```

Description

Applies an ACL to the VLAN identified by the current VLAN context.

The `no` form of this command removes application of the ACL from the VLAN identified by the current VLAN context.

Parameter	Description
ip ipv6 mac	Specifies the ACL type: ip for IPv4, ipv6 for IPv6, or mac for MAC ACL.
<ACL-NAME>	Specifies the ACL name.
in	Selects the inbound (ingress) traffic direction.

Usage

- Each ACL of a given type can be applied to the same VLAN once. Therefore, using the `apply access-list` command on a VLAN with an already-applied ACL of the same type, will replace the applied ACL.
- When an ACL is applied to a VLAN, it will create hardware entries on all line cards regardless of whether a VLAN member exists on any specific line card.

Examples

Applying My_ip_ACL to ingress traffic on VLAN range 20 to 25:

```
switch(config)# vlan 20-25
switch(config-vlan-<20-25>)# apply access-list ip My_ip_ACL in
```

Applying My_ip_ACL to ingress traffic on VLAN 10:

```
switch(config)# vlan 10
switch(config-vlan-10)# apply access-list ip My_ip_ACL in
```

Applying My_ipv6_ACL to ingress traffic on VLAN 10:

```
switch(config)# vlan 10
switch(config-vlan-10)# apply access-list ipv6 My_ipv6_ACL in
```

Applying My_mac_ACL to ingress traffic on VLAN 10:

```
switch(config)# vlan 10
switch(config-vlan-10)# apply access-list mac My_mac_ACL in
```

Replacing My_ipv6_ACL with My_Replacement_ACL on VLAN 10 (following the preceding examples):

```
switch(config)# vlan 10
switch(config-vlan-10)# apply access-list ipv6 My_Replacement_ACL in
```

Removing (unapplying) several ACLs on VLAN 10:

```
switch(config)# vlan 10
switch(config-vlan-10)# no apply access-list ipv6 My_Replacement_ACL in
switch(config-vlan-10)# no apply access-list mac My_mac_ACL in
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-vlan	Administrators or local user group members with execution rights for this command.

clear access-list hitcounts

```
clear access-list hitcounts { all | [{ip|ipv6|mac} <ACL-NAME>]
                             [interface <IF-NAME>| vlan <VLAN-ID>] [in|out|routed-in] }
```

Description

Clears the hit counts for ACLs with ACEs that include the `count` keyword.

Parameter	Description
all	Selects all ACLs.
ip ipv6 mac	Specifies the ACL type: ip for IPv4, ipv6 for IPv6, or mac for MAC.
<ACL-NAME>	Specifies the ACL name.
interface <IF-NAME>	Specifies the interface name (port or LAG).
vlan <VLAN-ID>	Specifies the VLAN.
in	Selects the inbound (ingress) traffic direction.
out	(not applicable to VLANs) selects the outbound (egress) traffic direction. Only for IPv4 ACLs applied to route-only ports. Not available for ACLs applied to either IPv4 bridged ports or IPv6 ports, or for MAC ACLs applied to ports or VLANs.
routed-in	Selects the routed traffic direction on which the ACL is applied. NOTE: This is only available for IPv4 and IPv6 ACLs applied to interface VLANs. routed-in selects the routed inbound (routed ingress) traffic direction.

Examples

Clearing the hit counts for My_ip_ACL applied to port 1/1/2 (egress):

```
switch# clear access-list hitcounts ip My_ip_ACL interface 1/1/2 out
```

Clearing the hit counts for My_ip_ACL applied to VLAN 10 (ingress):

```
switch# clear access-list hitcounts ip My_ip_ACL vlan 10 in
```

Clearing the hit counts for all ACLs:

```
switch# clear access-list hitcounts all
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

clear access-list hitcounts control-plane

```
clear access-list hitcounts [{ip|ipv6} <ACL-NAME>] control-plane vrf <VRF-NAME>
```

Description

Clears the hit counts for ACLs applied to the Control Plane VRF.

Parameter	Description
ip ipv6	Specifies the ACL type: ip for IPv4, or ipv6 for IPv6.
<ACL-NAME>	Specifies the ACL name.
vrf <VRF-NAME>	Specifies the VRF name.

Examples

Clearing the hit counts for an IPv4 ACL applied to the Control Plane default VRF:

```
switch# clear access-list hitcounts ip My_ipv4_ACL control-plane vrf default
```

Clearing the hit counts for an IPv6 ACL applied to the Control Plane default VRF:

```
switch# clear access-list hitcounts ipv6 My_ipv6_ACL control-plane vrf default
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

object-group address resequence

```
object-group {ip|ipv6} address <OBJECT-GROUP-NAME> resequence <STARTING-SEQUENCE-NUMBER> <INCREMENT>
```

Description

Reorders the sequence numbers in an address object group.

Parameter	Description
ip ipv6	Specifies the object group IP address type, either ip or ipv6.
<OBJECT-GROUP-NAME>	Specifies the address object group name.
<STARTING-SEQUENCE-NUMBER>	Specifies the starting sequence number.
<INCREMENT>	Specifies the sequence number increment.

Examples

Resequencing address object group my_ipv4_addr_group to use sequence numbers 5, 10, 15 and so on:

```
switch(config)# object-group address my_ipv4_addr_group resequence 5 5
switch(config)#
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

object-group address reset

```
object-group {ip|ipv6} address <OBJECT-GROUP-NAME> reset
```


Description

Resets the user configuration back to the active configuration. This command takes immediate effect, it is not saved in the user configuration. Use this command if misconfiguration of an address object group has occurred.

Parameter	Description
ip ipv6	Specifies the object group IP address type, either ip or ipv6.
<OBJECT-GROUP-NAME>	Specifies the address object group name.

Examples

Resetting IPv4 address object group my_ipv4_group:

```
switch(config)# object-group ip address my_ip_group reset
switch(config)#
```

Resetting IPv6 address object group my_ipv6_group:

```
switch(config)# object-group ipv6 address my_ipv6_group reset
switch(config)#
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

object-group all reset

```
object-group all reset
```

Description

Resets the user configuration back to the active configuration for all object types (address and port). This command takes immediate effect, it is not saved in the user configuration. Use this command if misconfiguration of address object groups and port object groups has occurred. Individual address and port object groups can be reset respectively with the `object-group address reset` and `object-group port reset` commands.

Examples

Resetting the user configuration for all object types (address and port):

```
switch(config)# object-group all reset
switch(config)#
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

object-group ip address

Syntax to create an IPv4 address object group and enter its context:

```
object-group ip address <OBJECT-GROUP-NAME>
```

```
no object-group ip address <OBJECT-GROUP-NAME>
```

Syntax (within the address object-group context) for creating or removing IPv4 address entries:

```
[<SEQUENCE-NUMBER>] <IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]
```

```
no <SEQUENCE-NUMBER>
```

Description

Creates an IPv4 address object group comprised of one or more address entries. Address groups are used solely as a shorthand way of specifying groups of addresses in the ACEs that make up ACLs. IPv4 address groups can be used only in the `access-list ip` command. Entering `object-group ip address` with an existing address group name, enables you to modify an existing address group.

The `no` form of this command deletes the entire address group or deletes a particular address group entry identified by sequence number.

Parameter	Description
<OBJECT-GROUP-NAME>	Specifies the address object group name.
<SEQUENCE-NUMBER>	Specifies a sequence number for the address entry. Range: 1 to 4294967295. When omitted, a sequence number 10 larger than the current highest sequence number is auto-assigned. Default auto-assigned sequence numbers are 10, 20, 30, and so on.
<IP-ADDRESS>[/{<PREFIX-LENGTH> <SUBNET-MASK>}]	Specifies the IPv4 address. <ul style="list-style-type: none"> ■ <IP-ADDRESS> - specifies the IPv4 host address. ■ <PREFIX-LENGTH> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 32.

Parameter	Description
	<ul style="list-style-type: none"> ▪ <code><SUBNET-MASK></code> - specifies the address bits to mask (dotted decimal notation).

Examples

Creating an IPv4 address group with two entries:

```
switch(config)# object-group ip address my_ipv4_addr_group
switch(config-addrgroup-ip)# 10 192.168.0.1
switch(config-addrgroup-ip)# 20 192.168.0.2
switch(config-addrgroup-ip)# exit
switch(config)# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
IPv4         my_ipv4_addr_group
           10 192.168.0.1
           20 192.168.0.2
```

Adding an entry to an existing IPv4 address group:

```
switch(config)# object-group ip address my_ipv4_addr_group
switch(config-addrgroup-ip)# 30 192.168.0.3
switch(config-addrgroup-ip)# exit
switch(config)# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
IPv4         my_ipv4_addr_group
           10 192.168.0.1
           20 192.168.0.2
           30 192.168.0.3
```

Removing an entry (20) from an existing IPv4 address group:

```
switch(config)# object-group ip address my_ipv4_addr_group
switch(config-addrgroup-ip)# no 20
switch(config-addrgroup-ip)# exit
switch(config)# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
IPv4         my_ipv4_addr_group
           10 192.168.0.1
           30 192.168.0.3
```

Removing an IPv4 address group:

```
switch(config)# no object-group ip address my_ipv4_addr_group
switch(config)# show object-group
No object group found.
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<pre>config</pre> <p>The <code>object-group ip address</code> command takes you into the named address group context (with prompt <code>switch (config-addrgroup-ip)#</code>) where you enter the addresses.</p>	Administrators or local user group members with execution rights for this command.

object-group ipv6 address

Syntax to create an IPv6 address object group and enter its context:

```
object-group ipv6 address <OBJECT-GROUP-NAME>
```

```
no object-group ipv6 address <OBJECT-GROUP-NAME>
```

Syntax (within the address object-group context) for creating or removing IPv6 address entries:

```
[ <SEQUENCE-NUMBER> ] <IP-ADDRESS> [ / { <PREFIX-LENGTH> | <SUBNET-MASK> } ]
```

```
no <SEQUENCE-NUMBER>
```

Description

Creates an IPv6 address object group comprised of one or more address entries. Address groups are used solely as a shorthand way of specifying groups of addresses in the ACEs that make up ACLs. IPv6 address groups can be used only in the `access-list ipv6` command. Entering `object-group ipv6 address` with an existing address group name, enables you to modify an existing address group.

The `no` form of this command deletes the entire address group or deletes a particular address group entry identified by sequence number.

Parameter	Description
<code><OBJECT-GROUP-NAME></code>	Specifies the address object group name.
<code><SEQUENCE-NUMBER></code>	Specifies a sequence number for the address entry. Range: 1 to 4294967295. When omitted, a sequence number 10 larger than the current highest sequence number is auto-assigned. Default auto-assigned sequence numbers are 10, 20, 30, and so on.
<code><IP-ADDRESS> [/ { <PREFIX-LENGTH> <SUBNET-MASK> }]</code>	<p>Specifies the IPv6 address.</p> <ul style="list-style-type: none"> ■ <code><IP-ADDRESS></code> - specifies the IPv6 host address. ○ <code><PREFIX-LENGTH></code> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 128.

Parameter	Description
	<ul style="list-style-type: none"> ◦ <i><SUBNET-MASK></i> - specifies the address bits to mask (dotted decimal notation).

Examples

Creating an IPv6 address group with two entries:

```
switch(config)# object-group ipv6 address my_ipv6_addr_group
switch(config-addrgroup-ipv6)# 10 1000::1
switch(config-addrgroup-ipv6)# 20 1000::2
switch(config-addrgroup-ipv6)# exit
switch(config)# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
IPv6         my_ipv6_addr_group
           10 1000::1
           20 1000::2
```

Adding an entry to an existing IPv6 address group:

```
switch(config)# object-group ipv6 address my_ipv6_addr_group
switch(config-addrgroup-ipv6)# 30 1000::3
switch(config-addrgroup-ipv6)# exit
switch(config)# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
IPv6         my_ipv6_addr_group
           10 1000::1
           20 1000::2
           30 1000::3
```

Removing an entry (20) from an existing IPv6 address group:

```
switch(config)# object-group ipv6 address my_ipv6_addr_group
switch(config-addrgroup-ipv6)# no 20
switch(config-addrgroup-ipv6)# exit
switch(config)# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
IPv6         my_ipv6_addr_group
           10 1000::1
           30 1000::3
```

Removing an IPv6 address group:

```
switch(config)# no object-group ipv6 address my_ipv6_addr_group
switch(config)# show object-group
No object group found.
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<pre>config The object-group ipv6 address command takes you into the named address group context (with prompt switch (config-addrgroup- ipv6) #) where you enter the addresses.</pre>	Administrators or local user group members with execution rights for this command.

object-group port

Syntax to create a Layer 4 port object group and enter its context:

```
object-group port <OBJECT-GROUP-NAME>
```

```
no object-group port <OBJECT-GROUP-NAME>
```

Syntax (within the port object-group context) for creating or removing Layer 4 port entries:

```
[<SEQUENCE-NUMBER>] { {eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT> }
```

```
no <SEQUENCE-NUMBER>
```

Description

Creates a Layer 4 port object group comprised of one or more port entries. Port groups are used solely as a shorthand way of specifying groups of ports in the ACEs that make up ACLs. Layer 4 port groups can be used only in the `access-list ip` and `access-list ipv6` commands. Entering `object-group port` with an existing port group name, enables you to modify an existing port group.

The `no` form of this command deletes the entire port group or deletes a particular port group entry identified by sequence number.

Parameter	Description
<code><OBJECT-GROUP-NAME></code>	Specifies the port object group name.
<code><SEQUENCE-NUMBER></code>	Specifies a sequence number for the port entry. Range: 1 to 4294967295. When omitted, a sequence number 10 larger than the current highest sequence number is auto-assigned. Default auto-assigned sequence numbers are 10, 20, 30, and so on.
<code>{ {eq gt lt} <PORT> range <MIN-PORT><MAX-PORT> }</code>	Specifies the port or port range. Port numbers are in the range of 0 to 65535. <ul style="list-style-type: none"> ■ <code>eq <PORT></code> - specifies the Layer 4 port. ■ <code>gt <PORT></code> - specifies any Layer 4 port greater than the indicated port.

Parameter	Description
	<ul style="list-style-type: none"> ■ <code>lt <PORT></code> - specifies any Layer 4 port less than the indicated port. ■ <code>range MIN-PORT> <MAX-PORT></code> - specifies the Layer 4 port range. <p>NOTE: When ACLs using ACEs defined with port groups are applied, the same number of hardware resources are consumed as when the ports are specified directly in the ACEs and not in a group. Keep this in mind when creating port groups that include many ports. Although hardware resource consumption is the same, with or without port groups used, it may not be immediately obvious that some port groups that you have defined, include many ports. It is recommended that you name port groups in a manner that reminds you that a group includes many ports.</p>

Examples

Creating a port group with two entries to cover port 80 plus ports 0 through 50:

```
switch(config)# object-group port my_port_group
switch(config-portgroup)# 10 eq 80
switch(config-portgroup)# 20 range 0 50
switch(config-portgroup)# exit
switch(config)# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
Port         my_port_group
           10 eq 80
           20 range 0 50
```

Adding an entry for ports greater than 65525 (covers ports 65526 through 65535):

```
switch(config)# object-group port my_port_group
switch(config-portgroup)# 30 gt 65525
switch(config-portgroup)# exit
switch(config)# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
Port         my_port_group
           10 eq 80
           20 range 0 50
           30 gt 65525
```

Removing an entry (#20) from the port group:

```

switch(config)# object-group port my_port_group
switch(config-portgroup)# no 20
switch(config-portgroup)# exit
switch(config)# show object-group
Type           Name
Sequence L4 Port(s)/IP Address
-----
Port          my_port_group
              10 eq 80
              30 gt 65525

```

Removing the port group:

```

switch(config)# no object-group port my_port_group
switch(config)# show object-group
No object group found.

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config The object-group ip port command takes you into the named port group context (with prompt switch(config-portgroup) #) where you specify the ports.	Administrators or local user group members with execution rights for this command.

object-group port resequence

object-group port <OBJECT-GROUP-NAME> resequence <STARTING-SEQUENCE-NUMBER> <INCREMENT>

Description

Reorders the sequence numbers in a port object group.

Parameter	Description
<OBJECT-GROUP-NAME>	Specifies the port object group name.
<STARTING-SEQUENCE-NUMBER>	Specifies the starting sequence number.
<INCREMENT>	Specifies the sequence number increment.

Examples

Resequencing port object group my_port_group to use sequence numbers 110, 120, 130 and so on:


```
switch(config)# object-group port my_port_group resequence 110 10
switch(config)#
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

object-group port reset

```
object-group port <OBJECT-GROUP-NAME> reset
```

Description

Resets the user configuration back to the active configuration. This command takes immediate effect, it is not saved in the user configuration. Use this command if misconfiguration of a port object group has occurred.

Parameter	Description
<OBJECT-GROUP-NAME>	Specifies the port object group name.

Examples

Resetting port object group my_port_group:

```
switch(config)# object-group port my_port_group reset
switch(config)#
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

show access-list

Syntax that filters by ACLs applied to an interface or VLAN:

```
show access-list [interface <IF-NAME>|vlan <VLAN-ID>] [ip|ipv6|mac]
[in|out|routed-in] [commands] [configuration] [vsx-peer]
```

Syntax that filters by the named ACL:

```
show access-list [ip|ipv6|mac] [<ACL-NAME>] [commands] [configuration] [vsx-peer]
```

Description

information about your defined ACLs and where they have been applied. When `show access-list` is entered without parameters, information for all ACLs is shown. The parameters filter the list of ACLs for which information is shown.

Available filtering includes:

- The content of a specific ACL.
- All ACLs of a specific type.
- All ACLs applied to a specific interface (port or split port or LAG).
- All ACLs applied to a specific VLAN.
- All ACLs applied in a particular direction.
- All IPv4 or IPv6 ACLs applied to interface VLANs (routed in).

Parameter	Description
<code>interface <IF-NAME></code>	Specifies the interface name (port or LAG).
<code>vlan <VLAN-ID></code>	Specifies the VLAN.
<code>ip ipv6 mac</code>	Specifies the ACL type: <ul style="list-style-type: none">■ <code>ip</code> for IPv4,■ <code>ipv6</code> for IPv6, or■ <code>mac</code> for MAC.
<code>in</code>	Selects the inbound (ingress) traffic direction.
<code>out</code>	(not applicable to VLANs) Selects the outbound (egress) traffic direction. Only for IPv4 ACLs applied to route-only ports. Not available for ACLs applied to either IPv4 bridged ports or IPv6 ports, or for MAC ACLs applied to ports or VLANs.
<code>routed-in</code>	Selects the routed traffic direction on which the ACL is applied. NOTE: This is only available for IPv4 and IPv6 ACLs applied to interface VLANs. <code>routed-in</code> selects the routed inbound (routed ingress) traffic direction.
<code><ACL-NAME></code>	Specifies the ACL name.
<code>commands</code>	Specifies that the ACL definition is to be shown as the commands and parameters used to create it rather than in tabular form.

Parameter	Description
configuration	Specifies that the user-configured ACLs be shown as entered, even if the ACLs are not active due to ACE-definition command issues or hardware issues. This parameter is useful if there is a mismatch between the entered configuration and the previous successfully programmed (active) ACLs configuration.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Creating an IPv4 ACL, applying it to an interface VLAN (routed in), and then showing ACL information filtered for that interface VAN:

```
switch(config)# access-list ip test
switch(config-acl-ip)# 10 permit any 1.1.1.2 2.2.2.2 count
switch(config-acl-ip)# 20 permit any 1.1.1.2 2.2.2.1 count
switch(config-acl-ip)# 30 permit any 2.2.2.2 1.1.1.2 count
switch(config-acl-ip)# 40 permit any 2.2.2.2 1.1.1.1 count
switch(config-acl-ip)# 50 permit any any any count
switch(config-acl-ip)# exit
switch(config)#
switch(config)# interface vlan100
switch(config-if-vlan)# apply access-list ip test routed-in
switch(config-if-vlan)# exit
switch(config)# show access-list interface vlan100 ip routed-in
```

Direction

Type	Name	Sequence	Comment	Ac	L3 Protocol	Source IP Address	Source L4 Port(s)	Destination IP Address	Destination L4 Port(s)	Additional Parameters

Routed Inbound

IPv4	test
10	permit any 1.1.1.2 2.2.2.2 Hit-counts: enabled
20	permit any 1.1.1.2 2.2.2.1 Hit-counts: enabled
30	permit any 2.2.2.2 1.1.1.2 Hit-counts: enabled
40	permit any 2.2.2.2 1.1.1.1 Hit-counts: enabled

```

50
  permit
  any
  any
  Hit-counts: enabled
-----

```

Showing an IPv4 ACL:

```

switch# show access-list ip MY_ACL
Type      Name
Sequence Comment
          Action                L3 Protocol
          Source IP Address     Source L4 Port(s)
          Destination IP Address Destination L4 Port(s)
          Additional Parameters
-----
IPv4      MY_ACL
10 permit
   any
   172.16.1.0/255.255.255.0      udp
20 permit
   172.16.2.0/255.255.0.0      tcp
   any                          > 1023
30 permit
   172.26.1.0//255.255.255.0   tcp
   any
   syn
   ack
   dscp 10
40 deny
   any
   any
   Hit-counts: enabled
-----

```

Showing an IPv4 ACL as commands:

```

switch# show access-list ip MY_ACL commands
access-list ip MY_ACL
 10 permit udp any 172.16.1.0/255.255.255.0
 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any
 30 permit tcp 172.26.1.0/255.255.255.0 any syn ack dscp 10
 40 deny any any any count

```

Showing IPv4 ACLs applied to VLAN 10, inbound:

```

switch# show access-list vlan 10 ip in
Type      Name
Sequence Comment
          Action                L3 Protocol
          Source IP Address     Source L4 Port(s)
          Destination IP Address Destination L4 Port(s)
          Additional Parameters
-----
IPv4      My_ip_ACL
10 permit
   any
   any
   Hit-counts: enabled
-----

```

```

    any
    172.16.1.0/255.255.255.0
20 permit 172.16.2.0/255.255.0.0 tcp
    > 1023
    any
30 permit 172.26.1.0//255.255.255.0 tcp
    any
    syn
    ack
    dscp 10
40 deny any any
    any
    any
Hit-counts: enabled
-----

```

Showing IPv6 ACLs applied to LAG 128, inbound:

```

switch# show access-list interface lag128 ipv6 in
Type      Name
Sequence  Comment
          Action
          Source IP Address
          Destination IP Address
          Additional Parameters
-----
IPv6      MY_IPV6_ACL
10 permit any 2001::1/64 udp
20 permit 2001:2001::2:1/128 tcp
    > 1023
30 permit 2001:2011::1/64 tcp
40 deny any any
    any
    any
Hit-counts: enabled
-----

```

Showing an IPv6 ACL as commands:

```

switch# show access-list ipv6 MY_IPV6_ACL commands
access-list ipv6 MY_IPV6_ACL
 10 permit udp any 2001::1/64
 20 permit tcp 2001:2001::2:1/128 gt 1023 any
 40 deny any any any count

```

Showing a MAC ACL:

```

switch# show access-list mac MY_MAC_ACL
Type      Name
Sequence  Comment
          Action
          Source MAC Address
          EtherType

```

```

Destination MAC Address
Additional Parameters
-----
MAC      MY_MAC_ACL
10 permit 1122.3344.5566/ffff.ffff.0000 any ipv6
    any
20 permit 1111.2222.3333 any
    aaaa.bbbb.cccc
    QoS Priority Code Point: 4
30 deny any any any any
    any
    Hit-counts: enabled
-----

```

Showing a MAC ACL as commands:

```

switch# show access-list mac MY_MAC_ACL commands
access-list mac MY_MAC_ACL
  10 permit 1122.3344.5566/ffff.ffff.0000 any ipv6
  20 permit aaaa.bbbb.cccc 1111.2222.3333 any pcp 4
  30 deny any any any count

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show access-list control-plane

```

show access-list [ip|ipv6] [<ACL-NAME>] control-plane [vrf <VRF-NAME>]
                  [commands] [configuration][vsx-peer]

```

Description

Shows information about your defined ACLs that have been applied to the Control Plane. When `show access-list control-plane` is entered without parameters, information for all ACLs applied to the Control Plane is shown. The parameters filter the list of ACLs for which information is shown.

Available filtering includes:

- The content of a specific ACL that has been applied to the Control Plane.
- All ACLs of a specific type that have been applied to the Control Plane.
- All ACLs applied to the Control Plane for a specific VRF.

Parameter	Description
ip ipv6	Specifies the ACL type: ip for IPv4, or ipv6 for IPv6.
<ACL-NAME>	Specifies the ACL name.
vrf <VRF-NAME>	Specifies the VRF name.
[commands]	Specifies that the ACL definition is to be shown as the commands and parameters used to create it rather than in tabular form.
[configuration]	Specifies that the user-configured ACLs be shown as entered, even if the ACLs are not active due to ACE-definition command issues or hardware issues. This parameter is useful if there is a mismatch between the entered configuration and the previous successfully programmed (active) ACLs configuration.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing an IPv4 ACL applied to the Control Plane default VRF:

```
switch# show access-list ip My_ipv4_ACL control-plane vrf default
Type      Name
Sequence  Action                               L3 Protocol
Source IP Address      Source L4 Port(s)
Destination IP Address Destination L4 Port(s)
Additional Parameters

-----
IPv4      My_ipv4_ACL
10 permit                               udp
   any
   172.16.1.0/24
20 permit                               tcp
   any                                   > 1023
   172.16.2.0/16
30 permit                               tcp
   any
   syn
   ack
   dscp 10
40 deny                                   any
   any
   any
Hit-counts: enabled
-----
```

Showing an IPv6 ACL applied to the Control Plane default VRF:

```
switch# show access-list ipv6 My_ipv6_ACL control-plane vrf default
Type      Name
Sequence  Action                               L3 Protocol
```

	Source IP Address	Destination IP Address	Additional Parameters	Source L4 Port(s)	Destination L4 Port(s)
IPv6	My_ipv6_ACL				
10	permit	any	2001::1/64	udp	
20	permit	any	2001:2001::2:1/128	tcp	> 1023
30	permit	any	2001:2011::1/64	tcp	
40	deny	any	any	any	
			Hit-counts: enabled		

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show access-list hitcounts

```
show access-list hitcounts { [{ip|ipv6|mac} <ACL-NAME>] [interface <IF-NAME> |
vlan <VLAN-ID>] [in|out|routed-in] [vsx-peer] }
```

Description

Shows the hit count of the number of times an ACL has matched a packet or frame for ACEs with the `count` keyword. For ACEs without the `count` keyword, a dash is shown in place of a hit count.

Parameter	Description
ip ipv6 mac	Specifies the ACL type: ip for IPv4, ipv6 for IPv6, or mac for MAC.
<ACL-NAME>	Specifies the ACL name.
interface <IF-NAME>	Specifies the interface name (port or split port or LAG).
vlan <VLAN-ID>	Specifies the VLAN.
in	Selects the inbound (ingress) traffic direction.

Parameter	Description
out	(Not applicable to VLANs) Selects the outbound (egress) traffic direction. Only for IPv4 ACLs applied to route-only ports. Not available for ACLs applied to either IPv4 bridged ports or IPv6 ports, or for MAC ACLs applied to ports or VLANs.
routed-in	Selects the routed traffic direction on which the ACL is applied. NOTE: This is only available for IPv4 and IPv6 ACLs applied to interface VLANs. routed-in selects the routed inbound (routed ingress) traffic direction.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Usage

- ACL hit counts are aggregated across all:
 - physical interfaces to which the ACL is applied to on ingress,
 - physical interfaces to which the ACL is applied to on egress,
 - VLANs to which the ACL is applied to on ingress.
 - Interface VLANs to which the IPv4 or IPv6 ACL is applied on routed ingress
- If an ACL with an ACE with the `count` keyword is applied to multiple physical interfaces or VLANs, the hit counts are aggregated. There is one aggregation for physical interfaces and another for VLANs.
- Accumulated hit counts for an applied ACL are cleared upon any modification of the ACL.

Examples

Showing the hit counts for My_ip_ACL applied to port 1/1/2:

```
switch# show access-list hitcounts ip My_ip_ACL interface 1/1/2
Statistics for ACL My_ip_ACL (ipv4):
interface 1/1/1-1/1/2,lag1 (out):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count
```

Showing the hit counts for My_ip_ACL applied to VLAN 10:

```
switch# show access-list hitcounts ip My_ip_ACL vlan 10
Statistics for ACL My_ip_ACL (ipv4):
vlan 10,20-100,300 (in):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count
```

Showing the hit counts for My_ip_ACL applied to interface VLAN 10:

```
switch# show access-list hitcounts ip My_ip_ACL vlan 10
Statistics for ACL My_ip_ACL (ipv4):
interface vlan 10,20,30 (routed-in):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count
```

Showing the hit counts for My_ip_ACL applied on any interface and direction:

```
switch# show access-list hitcounts ip My_ip_ACL vlan 10
switch# show access-list hitcounts ip My_ip_ACL
Statistics for ACL My_ip_ACL (ipv4):
interface 1/1/1 (in):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count

interface 1/1/1-1/1/2,lag1 (out):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count

interface 1/1/4.1,1/1/10.10 (in):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count

interface 1/1/4.1 (out):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count

interface vlan 10,20,30 (routed-in):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count

interface vlan 80-85 (routed-out):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
    0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
    - 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
    0 implicit deny any any any count

vlan 10,20-100,300 (in):
  Matched Packets Configuration
    - 10 permit udp any 172.16.1.0/255.255.255.0
```

```

0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
- 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
0 implicit deny any any any count

vlan 2-5 (out):
  Matched Packets Configuration
- 10 permit udp any 172.16.1.0/255.255.255.0
0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
- 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
0 implicit deny any any any count

vrf blue,default,red (control-plane):
  Matched Packets Configuration
- 10 permit udp any 172.16.1.0/255.255.255.0
0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
- 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
0 implicit deny any any any count

```

Command History

Release	Modification
10.07 or earlier	Updated command output to use interface and VLAN ranges to reflect aggregation.

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show access-list hitcounts control-plane

```
show access-list hitcounts [{ip|ipv6} <ACL-NAME>] control-plane vrf <VRF-NAME> [vsx-peer]
```

Description

Shows the hit count of the number of times an ACL (applied to the Control Plane) has matched a packet for ACEs with the `count` keyword. For ACEs without the `count` keyword, a dash is shown in place of a hit count.

Parameter	Description
<code>ip ipv6</code>	Specifies the ACL type: <code>ip</code> for IPv4, or <code>ipv6</code> for IPv6.
<code><ACL-NAME></code>	Specifies the ACL name.
<code>vrf <VRF-NAME></code>	Specifies the VRF name.
<code>vsx-peer</code>	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Usage

- ACL hit counts are aggregated across all VRFs to which the ACL is applied to on ingress.
- Accumulated hit counts for an applied ACL are cleared upon any modification of the ACL.

Examples

Showing the hit counts for an IPv4 ACL applied to the Control Plane default VRF:

```
switch# show access-list hitcounts ip My_ipv4_ACL control-plane vrf default
Statistics for ACL My_ip_ACL (ipv4):
vrf default (control-plane):
  Matched Packets  Configuration
- 10 permit udp any 172.16.1.0/255.255.255.0
0 20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any count
- 30 permit tcp 172.26.1.0/255.255.255.0 any dscp AF11 ack syn
0 implicit deny any any any count
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show capacities

show capacities <FEATURE> [vsx-peer]

Description

Shows system capacities and their values for all features or a specific feature.

Parameter	Description
<FEATURE>	Specifies a feature. For example, aaa or vrrp.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Usage

Capacities are expressed in user-understandable terms. Thus they may not map to a specific hardware or software resource or component. They are not intended to define a feature exhaustively.

Examples

Showing all available capacities for BGP:

```
switch# show capacities bgp

System Capacities: Filter BGP
Capacities Name                                     Value
-----
Maximum number of AS numbers in as-path attribute   32
...
```

Showing all available capacities for mirroring:

```
switch# show capacities mirroring

System Capacities: Filter Mirroring
Capacities Name                                     Value
-----
Maximum number of Mirror Sessions configurable in a system 4
Maximum number of enabled Mirror Sessions in a system      4
```

Showing all available capacities for MSTP:

```
switch# show capacities mstp

System Capacities: Filter MSTP
Capacities Name                                     Value
-----
Maximum number of mstp instances configurable in a system 64
```

Showing all available capacities for VLAN count:

```
switch# show capacities vlan-count

System Capacities: Filter VLAN Count
Capacities Name                                     Value
-----
Maximum number of VLANs supported in the system         4094
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Administrators or local user group members with execution rights for this command.

show capacities-status

show capacities-status <FEATURE> [vsx-peer]

Description

Shows system capacities status and their values for all features or a specific feature.

Parameter	Description
<FEATURE>	Specifies the feature, for example <code>aaa</code> or <code>vrrp</code> for which to display capacities, values, and status. Required.
<code>vsx-peer</code>	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing the system capacities status for all features:

```
switch# show capacities-status

System Capacities Status
Capacities Status Name                                     Value
Maximum
-----
Number of active gateway mac addresses in a system        0
  16
Number of aspath-lists configured                          0
  64
Number of community-lists configured                       0
  64
...
```

Showing the system capacities status for BGP:

```
switch# show capacities-status bgp

System Capacities Status: Filter BGP
Capacities Status Name                                     Value
Maximum
-----
Number of aspath-lists configured                          0
  64
Number of community-lists configured                       0
  64
Number of neighbors configured across all VRFs             0
  50
Number of peer groups configured across all VRFs           0
  25
Number of prefix-lists configured                          0
  64
Number of route-maps configured                           0
  64
Number of routes in BGP RIB                               0
```

256000

Number of route reflector clients configured across all VRFs

0

16

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Manager (#)	Administrators or local user group members with execution rights for this command.

show object-group

```
show object-group [{ip|ipv6} address | port] [<OBJECT-GROUP-NAME>] [commands]
[configuration]
```

Description

Shows information about your defined object groups. When `show object-group` is entered without parameters, information for all object groups is shown. The parameters filter the list of object groups for which information is shown.

Parameter	Description
[{ip ipv6} address port]	Specifies the object group type, either <code>address</code> for an IP address, or <code>port</code> .
<OBJECT-GROUP-NAME>	Specifies the object group name.
[commands]	Specifies that the object group definition is to be shown as the commands and parameters used to create it rather than in tabular form.
[configuration]	Specifies that the user-configured object groups be shown as configured. The output of the command with this parameter may not be the same as what is active on the switch due to a misconfigured object group. See <i>Examples</i> in this topic.
<code>vsx-peer</code>	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing configured object groups:

```

switch# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
IPv4          my_address_group
              10 192.168.0.1
              20 192.168.0.3
Port          my_port_group
              10 eq 80
              20 gt 65525
switch#
switch# show object-group commands
object-group ip address my_address_group
              10 192.168.0.1
              20 192.168.0.3
object-group port my_port_group
              10 eq 80
              20 gt 65525

```

Showing a misconfigured object group:

```

switch# show object-group
Type          Name
Sequence L4 Port(s)/IP Address
-----
! object-group ip address My_ip_object_group user configuration does not match
! the active hardware configuration. Run 'object-group ip address NAME reset'
! to reset the object group to match the active hardware configuration.
IPv4          my_address_group
switch#
switch# show object-group configuration
Type          Name
Sequence L4 Port(s)/IP Address
-----
! object-group ip address My_ip_object_group user configuration does not match
! the active hardware configuration. Run 'object-group ip address NAME reset'
! to reset the object group to match the active hardware configuration.
IPv4          my_address_group
              10 192.168.0.1
              20 192.168.0.3
switch#
switch# show object-group commands
! object-group ip address My_ip_object_group user configuration does not match
! the active hardware configuration. Run 'object-group ip address NAME reset'
! to reset the object group to match the active hardware configuration.
switch#
switch# show object-group commands configuration
! object-group ip address My_ip_object_group user configuration does not match
! the active hardware configuration. Run 'object-group ip address NAME reset'
! to reset the object group to match the active hardware configuration.
object-group ip address my_address_group
              10 192.168.0.1
              20 192.168.0.3

```

Resetting a misconfigured object group:

```

switch(config)# object-group all reset
switch(config)# exit
switch# show object-group

```



```

Type      Name
Sequence L4 Port(s)/IP Address
-----
IPv4      my_address_group
switch#
switch# show object-group configuration
Type      Name
Sequence L4 Port(s)/IP Address
-----
IPv4      my_address_group

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

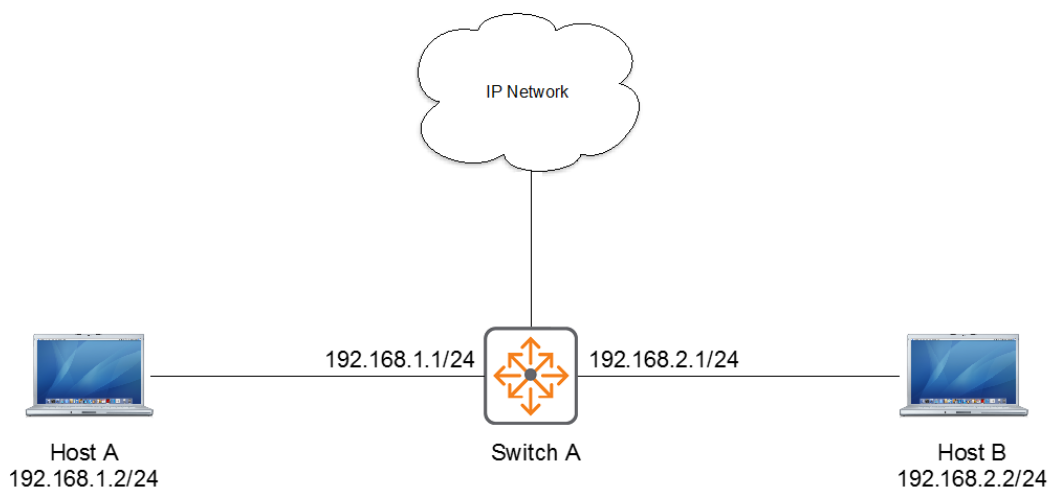
Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

This chapter shows examples for defining and applying IPv4 and IPv6 ACLs.

IPv4 ACL example overview

This example:

- Defines and applies an ACL to interface 1/1/1 on Switch A (see image in this topic) so that Host A is not able to send traffic to Host B, but it can communicate with all other devices in the network.
- Counts blocked packets.



Defining and applying an IPv4 ACL

Procedure

1. Begin defining an IPv4 ACL named FILTER_TO_HOST_B:

```
switch(config) # access-list ip FILTER_TO_HOST_B
```
2. Add an ACE that denies access from IP address 192.168.1.2 (Host A) to 192.168.2.2 (Host B):

```
switch(config-acl-ip) # deny any 192.168.1.2 192.168.2.2 log
```
3. Add an ACE that allows access from all other IP addresses:

```
switch(config-acl-ip) # permit any any any
```
4. Exit the ACL definition:

```
switch(config-acl-ip) # exit
```
5. Enter the context of the interface to which you will apply the ACL:

```
switch(config) # interface 1/1/1
```
6. Apply the FILTER_TO_HOST_B ACL to inbound (ingress) traffic:

```
switch(config-if)# apply access-list ip FILTER_TO_HOST_B in
```

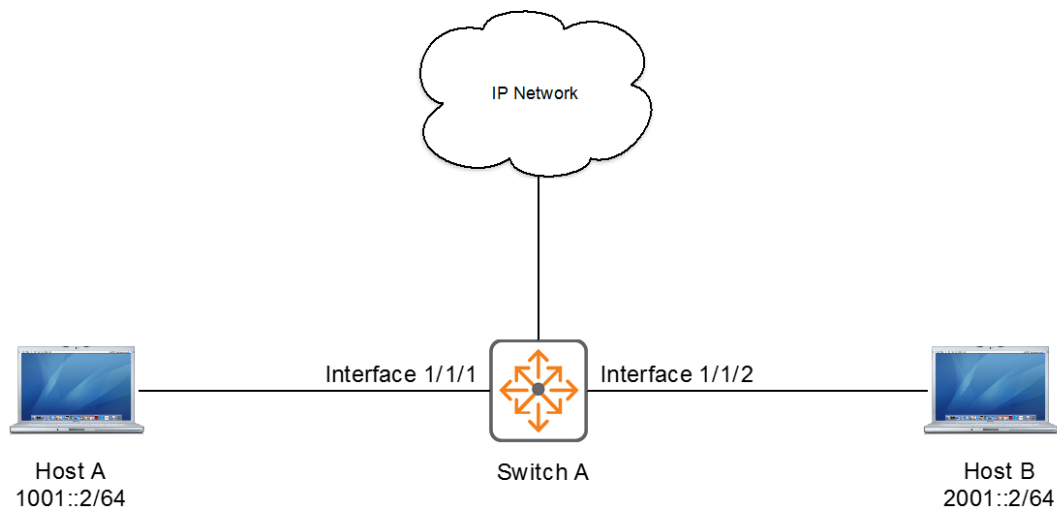
7. Show your ACL:

```
switch(config-if)# exit
switch# show access-list ip FILTER_TO_HOST_B
Type          Name
Sequence      Comment
Action
Source IP Address      L3 Protocol
Destination IP Address Source L4 Port(s)
Additional Parameters  Destination L4 Port(s)
-----
-
IPv4
10  FILTER_TO_HOST_B
   deny          any
   192.168.1.2
   192.168.2.2
   Logging: enabled
   Hit-counts: enabled
20  permit          any
   any
   any
-----
-
```

IPv6 ACL example overview

This example:

- Defines and applies an ACL to interface 1/1/1 on Switch A (see image in this topic) so that Host A is not able to send traffic to Host B, but it can communicate with all other devices in the network.
- Counts blocked packets.



Defining and applying an IPv6 ACL

Procedure

1. Begin defining an IPv6 ACL named V6_INPUT_FILTER:

```
switch(config)# access-list ipv6 V6_INPUT_FILTER
```
2. Add an ACE that denies access to an IP addresses 1001::2 through 2001::2 (includes Host B):

```
switch(config-acl-ipv6)# deny any 1001::2 2001::2 log
```
3. Add an ACE that allows access from all other IP addresses:

```
switch(config-acl-ipv6)# permit any any any
```
4. Exit the ACL definition:

```
switch(config-acl-ipv6)# exit
```
5. Enter the interface to which you will apply the ACL:

```
switch(config)# interface 1/1/1
```
6. Apply the V6_INPUT_FILTER ACL to inbound (ingress) traffic:

```
switch(config-if)# apply access-list ipv6 V6_INPUT_FILTER in
```
7. Show your ACL:

```
switch(config-if)# exit
switch# show access-list interface 1/1/1
Direction
Type      Name
  Sequence Comment
          Action
          Source IP Address
          Destination IP Address
          Additional Parameters
-----
Inbound
IPv6      V6_INPUT_FILTER
          10
          deny
          1001::2
          2001::2
          Logging: enabled
          Hit-counts: enabled
          20
          permit
          any
          any
-----
```

Classifier policies let a network administrator define sets of rules based on network traffic addressing or other header content, and use these rules to restrict or alter the passage of traffic through the switch. Choosing the rule criteria is called Classification, and one such rule, or list, is called a policy. Classification is achieved by creating a traffic class. The types of classes (MAC, IPv4, and IPv6) are each focused on relevant frame/packet characteristics. Classes can be configured to match or ignore almost any frame or packet header field. Network traffic passing through a switch can be classified based on many different frame/packet characteristics including, but not limited to:

- Frame ingress VLAN ID
- Source and/or destination Ethernet MAC, IPv4, or IPv6 address
- Layer 2 (EtherType) and Layer 3 (IP) protocol
- Layer 4 application ports

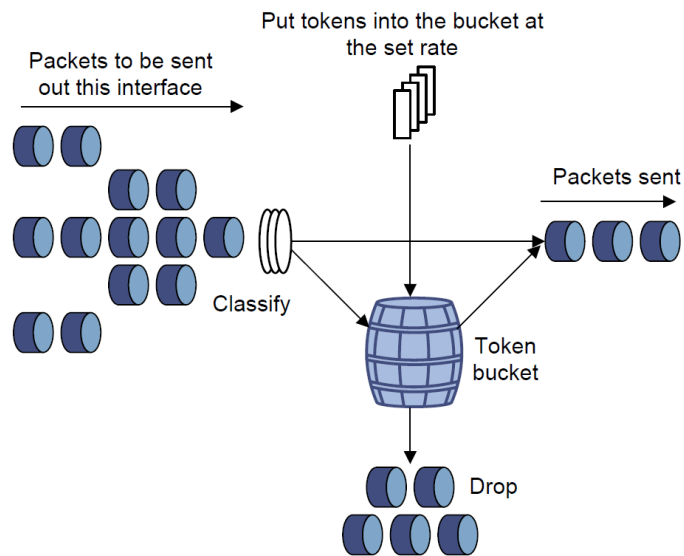
A policy contains one or more policy entries, which are listed according to priority by sequence number. A single policy entry contains a class and corresponding policy action. Policy action is taken on traffic matched by its corresponding class. A policy can be applied to an individual front plane port, a Link Aggregation Group (LAG) interface, or a VLAN.



See also [ACL and Policy hardware resource considerations](#).

Traffic policing

Traffic policing supports policing of the inbound traffic. A typical application of traffic policing is to supervise the specification of traffic entering a network and limit it within a reasonable range. Another application is to "discipline" the extra traffic to prevent aggressive use of network resources by an application. For example, you can limit bandwidth for HTTP packets to less than 50% of the total. If the traffic of a session exceeds the limit, traffic policing can drop the packets or reset the IP precedence of the packets. In the following illustrated example, outbound traffic is policed:



Traffic policing is widely used in policing traffic entering the ISP networks. It can classify the policed traffic and take predefined policing actions on each packet depending on the evaluation result:

- Forwarding the packet if the evaluation result is "conforming."
- Dropping the packet if the evaluation result is "excess."
- Forwarding the packet with its precedence remarked when the evaluation result is "conforming."

Types of policy actions

The policy actions are broadly classified in the following categories:

- Remark actions
- Police actions
- Other actions

Each policy entry can have a combination of policy actions from these multiple categories, which are executed in the order of configuration.

Remark actions

This category contains the following actions:

- **Priority Code Point (PCP):** 3-bit field in layer 2 802.1Q header refers to a class of service and maps to a frame priority level.
- **IP precedence:** 3-bit field in IP header which denotes the importance or priority of the datagram.
- **IP Differentiated Services Code Point (DSCP):** 6-bit field in IP header for packet classification.
- **Local Priority:** Change the internal priority used to queue the packets for transmission. Local priority can be used to rewrite the priority of traffic classes local to the system based on the QoS mapping settings without changing the IP header or the 802.1Q header. Remark actions other than local priority only change packets as they leave the switch. The local priority action can be combined with the other remark actions to remark packets and change the internal priority to reflect the new priority.

Police actions

Traffic policing meters inbound traffic on an interface or VLAN based on the following traffic parameters:

- Committed information rate (CIR): Bandwidth limit for guaranteed traffic.
- Committed burst size (CBS): Maximum packet size permitted for bursts of data that exceed the CIR.

Based on these parameters, packets are dropped when traffic exceeds the bandwidth limit (CIR) and the burst size for guaranteed traffic (CBS).

Other actions

Other actions include Drop: Drop the packet, and Mirror: Mirror the packets to a specified mirroring session. For details, see the *Monitoring Guide*.

How policy matching works

A policy can be applied to an interface or VLAN to affect/control traffic arriving on that interface or VLAN (inbound (ingress)). A single policy entry matches on one or more characteristics of the particular traffic type and has a configured action to continue through the switch. This matching occurs by beginning with the entry with the lowest sequence number. The entry is then compared against the incoming frame to its particular match characteristics. If there is a match, the action is taken.

If there is no match, the match characteristics of the next sequence are compared to the relevant frame/packet details. If there is a match, the specified actions are taken. This process continues until a match is found; otherwise, the packet is permitted to flow through the switch unaltered. The "implicit permit" behavior of policy matching differs from the "implicit deny" behavior of ACL matching.

Active class configuration versus user-specified configuration

The output of the `show class` command displays the active class configurations. Active class configurations are the classes that have been configured and accepted by the system.

The output of the `show class` command with the `configuration` parameter, displays the classes that have been configured by the user.

Discrepancies might occur between the active class configurations and the user-specified configurations. In the user-specified class configurations, unsupported command parameters may have been configured, or class can be modified after policy application and may have been unsuccessful due to lack of hardware resources.

To determine if a discrepancy exists between what was configured and what is active, run any variant of the `show class` command. If the active classes and configured classes are not the same, a warning message is displayed.

```
! class MY_CLASS user configuration does not match active configuration.  
! run 'class TYPE NAME reset' to reset class to match active configuration.
```

If the configured class is processing and you entered the `show class` command with parameters, the following in-progress message is displayed:

```
! class ip MY_CLASS user configuration currently being processed  
! run 'class TYPE NAME reset' to reset class to match active configuration.
```

If the configured class is processing and you entered the `show class` command without parameters, the following in-progress message is displayed:

```
% Warning: MY_CLASS user configuration currently being processed
% run 'class TYPE NAME reset' to reset class to match active configuration.
```

If the warning message or in-progress message is displayed, additional changes may be made until the error message is no longer displayed. Or you can use the `class {all|ip <class-name>|ipv6 <class-name>|mac <class-name>} reset` command to change the user-specified configuration to match the active configuration.



The `show running-config` command also shows a warning about classes that are in progress or failed.

Example

Resetting the user-specified class configuration to the active configuration:

```
switch(config)# class all reset
```

Active policy configuration versus user-specified configuration

The output of the `show policy` command displays the active policy configurations. Active policy configurations are the policies that have been configured and accepted by the system. With applied policies, the active configuration displays the interfaces on which the policies have successfully been programmed in hardware.

The output of the `show policy` command with the `configuration` parameter, displays the policies that have been configured by the user.

Discrepancies might exist between the active policy configurations and the user-specified configurations. In the user-specified policy configurations, unsupported command parameters might have been configured, or an application of a policy might have been unsuccessful because of a lack of hardware resources.

To determine if a discrepancy exists between the configuration and what is active, run any variant of the `show policy` command. If the active policies and configured policies are not the same, a warning message is displayed in the output of the `show` command.

```
! policy MY_POLICY user configuration does not match active configuration.
! run 'policy NAME reset' to reset policy to match active configuration.
```

The switch displays an `in progress` message while it processes the configured policy:

```
! policy MY_POLICY user configuration currently being processed
! run 'policy NAME reset' to reset policy to match active configuration.
```

If the warning message or in progress message is displayed, additional changes may be made until the error message is no longer displayed. Or you can use the `policy <policy-name> reset` command to change the user-specified configuration to match the active configuration.

Example

Resetting `MY_POLICY`:


```
switch(config)# policy MY_POLICY reset
```

Considerations for when a policy is applied per interface



This section is only applicable to polices applied to physical interfaces and LAGs using the `per-interface` parameter.

The `reset` command (mentioned in the previous section) is not useful if one or more unique instances of a policy created using the `per-interface` parameter fail to update in hardware even though the parent policy does update. If this occurs, you can make additional changes to the policy and its applications to correct the discrepancy until the error messages are no longer displayed. Alternatively consider using command `checkpoint-rollback` as described in the *AOS-CX Fundamentals Guide*.

Policies using the `per-interface` parameter have slightly different warning and in-progress messages due to unique instances of the policy being created and applied to individual physical interfaces and LAGs.

For example, this is how the warning messages will appear if the unique instances of the policy for interfaces 1/1/2-1/1/3 fail to update while the unique instances of the policy for interfaces 1/1/1,1/1/4 successfully update.

```
switch(config)# show policy commands
! policy my_policy user configuration does not match active configuration on
interface 1/1/2 for ingress.
! policy my_policy user configuration does not match active configuration on
interface 1/1/3 for ingress.
policy my_policy
  10 class ip my_ip_class action drop
interface 1/1/1
  apply policy my_policy in per-interface
! policy my_policy user configuration does not match active configuration.
interface 1/1/2
  apply policy my_policy in per-interface
! policy my_policy user configuration does not match active configuration.
interface 1/1/3
  apply policy my_policy in per-interface
interface 1/1/4
  apply policy my_policy in per-interface

switch(config)# show policy
      Name
Sequence Comment
      Class Type
              action
-----
% Warning: my_policy user configuration does not match active configuration on
interface 1/1/2 for ingress.
% Warning: my_policy user configuration does not match active configuration on
interface 1/1/3 for ingress.
      my_policy
10
      my_ip_class ipv4
              drop
```

This is how the in-progress messages will appear if the child policies for interfaces 1/1/2-1/1/3 are currently updating while the child policies for interfaces 1/1/1,1/1/4 have successfully updated.

```

switch(config)# show policy commands
! policy my_policy user configuration currently being processed on interface 1/1/2
for ingress.
! policy my_policy user configuration currently being processed on interface 1/1/3
for ingress.
! run 'show policy [commands]' to display active policy configuration.
policy my_policy
  10 class ip my_ip_class action drop
interface 1/1/1
  apply policy my_policy in per-interface
! policy my_policy user configuration currently being processed
! run 'show policy [commands]' to display active policy configuration.
interface 1/1/2
  apply policy my_policy in per-interface
! policy my_policy user configuration currently being processed
! run 'show policy [commands]' to display active policy configuration.
interface 1/1/3
  apply policy my_policy in per-interface
interface 1/1/4
  apply policy my_policy in per-interface

switch(config)# show policy
      Name
Sequence Comment
      Class Type
              action
-----
% Warning: my_policy user configuration currently being processed on interface 1/1/2
for ingress.
% Warning: my_policy user configuration currently being processed on interface 1/1/3
for ingress.
%      run 'show policy [commands]' to display active policy configuration.
      my_policy
      10
      my_ip_class ipv4
              drop

```

This is how the warning messages will appear if the child policies for interfaces 1/1/2-1/1/3 failed to apply or replace while the child policies for interfaces 1/1/1,1/1/4 have successfully applied or replaced.

```

switch(config)# show policy commands
policy my_policy
  10 class ip my_ip_class action drop
interface 1/1/1
  apply policy my_policy in per-interface
! policy my_policy user configuration does not match active configuration.
! run 'policy NAME reset' to reset policy to match active configuration.
interface 1/1/2
  apply policy my_policy in per-interface
! policy my_policy user configuration does not match active configuration.
! run 'policy NAME reset' to reset policy to match active configuration.
interface 1/1/3
  apply policy my_policy in per-interface
interface 1/1/4
  apply policy my_policy in per-interface

switch(config)# show policy
      Name
Sequence Comment
      Class Type
              action

```

```

-----
my_policy
10
  my_ip_class ipv4
    drop

```

Classifier policy commands

Classifier policy application

Classifier policies can be applied as follows ("Rt-In" = "Routed-In"):

Policy type Direction	IPv4 In	IPv4 Rt-In	IPv6 In	IPv6 Rt-In	IPv4+6 In	IPv4+6 Rt-In	MAC In
L2 interface (port)	Yes		Yes		Yes		Yes
L2 LAG	Yes		Yes		Yes		Yes
L3 interface (port)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
L3 LAG	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VLAN	Yes		Yes		Yes		Yes
Interface VLAN		Yes (PBR)		Yes (PBR)		Yes (PBR)	

The following match criteria are not supported. If any of these match criteria are attempted to be configured, an error message will be displayed and the action will not be completed.



```

TCP flags CWR and ECE
TCP flags and TTL (hop limit) on IP classes
Fragment on IPv6 classes in VLAN policies
VLAN ID in VLAN policies

```

apply policy (config-if, config-lag-if, config-if-vlan, config-vlan)

Context config-if, config-lag-if:

```

apply policy <POLICY-NAME> {in|out|routed-in} [per-interface]
no apply policy <POLICY-NAME> {in|out|routed-in} [per-interface]

```

Context config-vlan:

```

apply policy <POLICY-NAME> {in|out}
no apply policy <POLICY-NAME> {in|out}

```

Context config-if-vlan:

```

apply policy <POLICY-NAME> routed-in
no apply policy <POLICY-NAME> routed-in

```

Description

Applies a policy to the current physical interface port or LAG or VLAN context.

Only one direction of a policy can be applied to an interface or VLAN at a time, thus using the `apply` command on an interface or VLAN with an already-applied policy of the same direction will replace the currently applied policy.



The VLAN context supports the `in` and `out` directions, which apply to both bridged and routed traffic. The Interface VLAN context only supports the `routed-in` direction which applies only to routed traffic.

The `no` form of this command removes a policy from the interface or VLAN specified by the current context.

Parameter	Description
<code><POLICY-NAME></code>	Specifies the policy to apply.
<code>in</code>	Selects the inbound (ingress) traffic direction.
<code>out</code>	Selects the outbound (egress) traffic direction.
<code>routed-in</code>	Selects routed in traffic.
<code>per-interface</code>	Specifies that unique instances of the policy be applied to each interface or LAG rather than the default of sharing the policy across all interfaces and LAGs.

Usage (applies to `config-if`, `config-lag-if` contexts)

- When `per-interface` is included, unique instances of the policy are applied to each physical interface port or LAG rather than the default of sharing the policy across all interfaces and LAGs. The unique instance of a policy has a parent-child relationship with the policy from which it was created. The `per-interface` option is useful when you want unique policers to be created for each interface or LAG rather than using shared policers. It is also useful when you want the statistics (hit counts and conform rate) to be specific to an interface or LAG rather than being aggregated. Because `per-interface` creates more hardware instances of a policy, resource consumption may increase significantly. It is recommended that you use `show resources` to monitor resource utilization as configuration is applied.

Usage (applies to `config-vlan` context)

- Only one policy type may be applied to a VLAN at a time. Therefore, using the `apply policy` command on a VLAN with an already-applied policy of the same type, will replace the applied policy.
- When a policy is applied to a VLAN, it will create hardware entries on all line cards regardless of whether a VLAN member exists on any specific line card.

Examples

Applying a policy to an interface (ingress):

```
switch(config)# interface 1/1/1
switch(config-if)# apply policy MY_POLICY1 in
```

Applying a policy to an interface (ingress) specifying `per-interface`:

```
switch(config)# interface 1/1/2
switch(config-if)# apply policy MY_POLICY1 in per-interface
```

Applying a policy to an interface (egress):

```
switch(config)# interface 1/1/2
switch(config-if)# apply policy MY_POLICY2 out
```

Applying a policy to an interface (egress) specifying per-interface:

```
switch(config)# interface 1/1/2
switch(config-if)# apply policy MY_POLICY2 out per-interface
```

Applying a policy to an interface range (ingress):

```
switch(config)# interface 1/1/3-1/1/6
switch(config-if-<1/1/2-1/1/5>)# apply policy MY_POLICY3 in
```

Applying a policy to an interface range (ingress) specifying per-interface:

```
switch(config)# interface 1/1/7-1/1/9
switch(config-if-<1/1/2-1/1/5>)# apply policy MY_POLICY4 in per-interface
```

Removing a policy from an interface (ingress):

```
switch(config)# interface 1/1/1
switch(config-if)# no apply policy MY_POLICY1 in
```

Removing a policy from an interface range (ingress):

```
switch(config)# interface 1/1/3-1/1/6
switch(config-if-<1/1/3-1/1/6>)# no apply policy MY_POLICY3 in
```

Applying a policy to a LAG (ingress):

```
switch(config)# interface lag 100
switch(config-lag-if)# apply policy MY_POLICY5 in
```

Applying a policy to a LAG (ingress) specifying per-interface:

```
switch(config)# interface lag 200
switch(config-lag-if)# apply policy MY_POLICY5 in per-interface
```

Removing a policy from a LAG (ingress):

```
switch(config)# interface lag 100
switch(config-lag-if)# no apply policy MY_POLICY5 in
```

Applying a policy to a VLAN (ingress):

```
switch(config)# vlan 1  
switch(config-vlan)# apply policy MY_POLICY6 in
```

Applying a policy to multiple VLANs (ingress):

```
switch(config)# vlan 10,20  
switch(config-vlan-<10,20>)# apply policy MY_POLICY7 in
```

Applying a policy to an interface VLAN routed (ingress):

```
switch(config)# vlan 1  
switch(config-if-vlan)# apply policy MY_POLICY8 routed-in
```

Applying a policy to an interface VLAN range routed (ingress):

```
switch(config)# vlan 2-5  
switch(config-if-vlan-<2-5>)# apply policy MY_POLICY8 routed-in
```

Removing a policy from a VLAN (ingress):

```
switch(config)# vlan 1  
switch(config-vlan)# no apply policy MY_POLICY6 in
```

Removing a policy from multiple VLANs (ingress):

```
switch(config)# vlan 10,20  
switch(config-vlan-<10,20>)# no apply policy MY_POLICY7 in
```

Removing a policy from an interface VLAN routed (ingress):

```
switch(config)# vlan 1  
switch(config-if-vlan)# no apply policy MY_POLICY8 routed-in
```

Removing a policy from an interface VLAN range routed (ingress):

```
switch(config)# vlan 2-5  
switch(config-if-vlan-<2-5>)# no apply policy MY_POLICY8 routed-in
```

Command History

Release	Modification
10.08	Added [per-interface] parameter. Updated examples.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config-if config-lag-if config-vlan config-if-vlan	Administrators or local user group members with execution rights for this command.

class copy

```
class {ip|ipv6|mac} <CLASS-NAME> copy <DESTINATION-CLASS>
```

Description

Copies a class to a new destination class or overwrites an existing class. Copying a class copies all entries as well.

Parameter	Description
{ip ipv6 mac} <CLASS-NAME>	Specifies the type and name of the class to be copied.
<DESTINATION-CLASS>	Specifies the name of the destination class.

Examples

Copying an IPv4 class. Copying a class with entries copies all its entries as well:

```
switch(config)# class ip MY_IP_CLASS copy MY_IP_CLASS2
switch(config)# do show class
Type      Name
Sequence Comment
Action
Source IP Address      L3 Protocol
Destination IP Address Source L4 Port(s)
Additional Parameters  Destination L4 Port(s)
-----
IPv4      MY_IP_CLASS
11 ignore                udp
   any
   any
21 match                tcp
   192.168.0.1
   192.168.0.2
-----
IPv4      MY_IP_CLASS2
11 ignore                udp
   any
   any
21 match                tcp
   192.168.0.1
   192.168.0.2
```

Copying an IPv6 class:

```
switch(config)# class ipv6 MY_IPV6_CLASS copy MY_IPV6_CLASS2
switch(config)# do show class
Type      Name
Sequence Comment
```

	Action		L3 Protocol
	Source IP Address		Source L4 Port(s)
	Destination IP Address		Destination L4 Port(s)
	Additional Parameters		
IPv6	MY_IPV6_CLASS		
	2 ignore		udp
	any		
	any		
IPv6	MY_IPV6_CLASS2		
	2 ignore		udp
	any		
	any		

Copying a MAC class:

```
switch(config)# class mac MY_MAC_CLASS copy MY_MAC_CLASS2
switch(config)# do show class
```

Type	Name		
Sequence	Comment		EtherType
	Action		
	Source MAC Address		
	Destination MAC Address		
	Additional Parameters		
MAC	MY_MAC_CLASS		
	2 ignore		arp
	any		
	any		
MAC	MY_MAC_CLASS2		
	2 ignore		arp
	any		
	any		

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

class ip

Syntax to create an IPv4 class and enter its context. Plus syntax to remove a class:

```
class ip <CLASS-NAME>
```

```
no class ip <CLASS-NAME>
```

Syntax (within the class context) for creating or removing class entries for protocols ah, gre, esp, igmp, ospf, pim (ip is available as an alias for any):


```

[<SEQUENCE-NUMBER>]
{match|ignore}
{any|ip|ah|gre|esp|igmp|ospf|pim|<IP-PROTOCOL-NUM>}
{any|<SRC-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}]}
{any|<DST-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count]

no <SEQUENCE-NUMBER>

```

Syntax (within the class context) for creating or removing class entries for protocols sctp, tcp, udp:

```

[<SEQUENCE-NUMBER>]
{match|ignore}
{sctp|tcp|udp}
{any|<SRC-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>]
{any|<DST-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>]
[urg] [ack] [psh] [rst] [syn] [fin] [established]
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count]

no <SEQUENCE-NUMBER>

```

Syntax (within the class context) for creating or removing class entries for protocol icmp:

```

[<SEQUENCE-NUMBER>]
{match|ignore}
{icmp}
{any|<SRC-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}]}
{any|<DST-IP-ADDRESS>[/ {<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[icmp-type {echo|echo-reply|<ICMP-TYPE-VALUE>}] [icmp-code <ICMP-CODE-VALUE>]
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count]

no <SEQUENCE-NUMBER>

```

Syntax (within the class context) for class entry comments:

```

[<SEQUENCE-NUMBER>] comment <TEXT-STRING>

no <SEQUENCE-NUMBER> comment

```

Description

Creates or modifies an IPv4 traffic class to match specified packets. Class is composed of one or more class entries ordered and prioritized by sequence numbers. With this command, the class can classify traffic based on IPv4 header information.

The `no` form of the command can be used to delete either an IPv4 traffic class (use `no` with the class command) or an individual IPv4 traffic class entry (use `no` with the sequence number).

Parameter	Description
ip	Specifies create or modify an IPv4 class.
<CLASS-NAME>	Specifies the name of this class.
<SEQUENCE-NUMBER>	Specifies a sequence number for the class entry. Optional. Range: 1-4294967295.

Parameter	Description
{match ignore}	Creates a rule to match or ignore specified packets.
<IP-PROTOCOL-NUM>	Specifies the protocol as its Internet Protocol number. For example, 2 corresponds to the IGMP protocol. Range: 0 to 255.
{any <SRC-IP-ADDRESS>[/{<PREFIX-LENGTH> <SUBNET-MASK>}]}	Specifies the source IPv4 address. <ul style="list-style-type: none"> ■ any - specifies any source IPv4 address. ■ <SRC-IP-ADDRESS> - specifies the source IPv4 host address. <ul style="list-style-type: none"> ○ <PREFIX-LENGTH> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 32. ○ <SUBNET-MASK> - specifies the address bits to mask (dotted decimal notation).
{any <DST-IP-ADDRESS>[/{<PREFIX-LENGTH> <SUBNET-MASK>}]}	Specifies the destination IPv4 address. <ul style="list-style-type: none"> ■ any - specifies any destination IPv4 address. ■ <DST-IP-ADDRESS> - specifies the destination IPv4 host address. <ul style="list-style-type: none"> ○ <PREFIX-LENGTH> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 32. ○ <SUBNET-MASK> - specifies the address bits to mask (dotted decimal notation).
[{eq gt lt} <PORT> range <MIN-PORT><MAX-PORT>]	Specifies the port or port range. Port numbers are in the range of 0 to 65535. <ul style="list-style-type: none"> ■ eq <PORT> - specifies the Layer 4 port. ■ gt <PORT> - specifies any Layer 4 port greater than the indicated port. ■ lt <PORT> - specifies any Layer 4 port less than the indicated port. ■ range <MIN-PORT> <MAX-PORT> - specifies the Layer 4 port range.
urg	Specifies matching on the TCP Flag: Urgent.

Parameter	Description
ack	Specifies matching on the TCP Flag: Acknowledgment.
psh	Specifies matching on the TCP Flag: Push buffered data to receiving application.
rst	Specifies matching on the TCP Flag: Reset the connection.
syn	Specifies matching on the TCP Flag: Synchronize sequence numbers.
fin	Specifies matching on the TCP Flag: Finish connection.
established	Specifies matching on the TCP Flag: Established connection.
dscp <DSCP-SPECIFIER>	<p>Specifies the Differentiated Services Code Point (DSCP), either a numeric <DSCP-VALUE> (0 to 63) or one of these keywords:</p> <ul style="list-style-type: none"> ■ AF11 - DSCP 10 (Assured Forwarding Class 1, low drop probability) ■ AF12 - DSCP 12 (Assured Forwarding Class 1, medium drop probability) ■ AF13 - DSCP 14 (Assured Forwarding Class 1, high drop probability) ■ AF21 - DSCP 18 (Assured Forwarding Class 2, low drop probability) ■ AF22 - DSCP 20 (Assured Forwarding Class 2, medium drop probability) ■ AF23 - DSCP 22 (Assured Forwarding Class 2, high drop probability) ■ AF31 - DSCP 26 (Assured Forwarding Class 3, low drop probability) ■ AF32 - DSCP 28 (Assured Forwarding Class 3, medium drop probability) ■ AF33 - DSCP 30 (Assured Forwarding Class 3, high drop probability) ■ AF41 - DSCP 34 (Assured

Parameter	Description
	<p>Forwarding Class 4, low drop probability)</p> <ul style="list-style-type: none"> ■ AF42 - DSCP 36 (Assured Forwarding Class 4, medium drop probability) ■ AF43 - DSCP 38 (Assured Forwarding Class 4, high drop probability) ■ CS0 - DSCP 0 (Class Selector 0: Default) ■ CS1 - DSCP 8 (Class Selector 1: Scavenger) ■ CS2 - DSCP 16 (Class Selector 2: OAM) ■ CS3 - DSCP 24 (Class Selector 3: Signaling) ■ CS4 - DSCP 32 (Class Selector 4: Realtime) ■ CS5 - DSCP 40 (Class Selector 5: Broadcast video) ■ CS6 - DSCP 48 (Class Selector 6: Network control) ■ CS7 - DSCP 56 (Class Selector 7) ■ EF - DSCP 46 (Expedited Forwarding)
ecn <ECN-VALUE>	Specifies an Explicit Congestion Notification value. Range: 0 to 3.
ip-precedence <IP-PRECEDENCE-VALUE>	Specifies an IP precedence value. Range: 0 to 7.
tos <TOS-VALUE>	Specifies the Type of Service value. Range: 0 to 31.
fragment	Specifies a fragment packet.
vlan <VLAN-ID>	<p>Specifies VLAN tag to match on. 802.1Q VLAN ID.</p> <p>NOTE: This parameter cannot be used in any class that will be applied to a VLAN.</p>
ttl <TTL-VALUE>	Specifies a time-to-live (hop limit) value. Range: 0 to 255.
count	Keeps the hit counts of the number of packets matching this class entry.

Parameter	Description
[<SEQUENCE-NUMBER>] comment <TEXT-STRING>	Adds a comment to a class entry. The <code>no</code> form removes only the comment from the class entry.

Usage

- Entering an existing <CLASS-NAME> value will cause the existing class to be modified, with any new <SEQUENCE-NUMBER> value creating an additional class entry, and any existing <SEQUENCE-NUMBER> value replacing the existing class entry with the same sequence number.
- If no sequence number is specified, a new class entry will be appended to the end of the class with a sequence number equal to the highest class entry currently in the list plus 10.
- If the <IP-PROTOCOL-NUM> parameter is used instead of a protocol name, ensure that any needed class entry-definition parameters specific to the selected protocol are also provided.

Examples

Creating an IPv4 class with three entries:

```
switch(config)# class ip MY_IP_CLASS
switch(config-class-ip)# 10 match icmp any any
switch(config-class-ip)# 20 ignore udp any any
switch(config-class-ip)# 30 match tcp 192.168.0.1 192.168.0.2
switch(config-class-ip)# exit

switch(config)# do show class
Type          Name
  Sequence Comment
           Action                L3 Protocol
           Source IP Address      Source L4 Port(s)
           Destination IP Address  Destination L4 Port(s)
           Additional Parameters
-----
IPv4          MY_IP_CLASS
  10 match
    any
    any
  20 ignore
    any
    any
  30 match
    192.168.0.1
    192.168.0.2
    tcp
```

Adding a comment to an existing IPv4 class entry:

```
switch(config)# class ip MY_IP_CLASS
switch(config-class-ip)# 30 comment myipClass
switch(config-class-ip)# exit

switch(config)# do show class
Type          Name
  Sequence Comment
           Action                L3 Protocol
           Source IP Address      Source L4 Port(s)
           Destination IP Address  Destination L4 Port(s)
```

```

Additional Parameters
-----
IPv4      MY_IP_CLASS
10 match          icmp
   any
   any
20 ignore        udp
   any
   any
30 myipClass
   match          tcp
   192.168.0.1
   192.168.0.2

```

Removing a comment from an existing IPv4 class entry:

```

switch(config)# class ip MY_IP_CLASS
switch(config-class-ip)# no 30 comment
switch(config-class-ip)# exit

switch(config)# do show class
Type      Name
Sequence Comment
Action    L3 Protocol
Source IP Address Source L4 Port(s)
Destination IP Address Destination L4 Port(s)
Additional Parameters
-----
IPv4      MY_IP_CLASS
10 match          icmp
   any
   any
20 ignore        udp
   any
   any
30 match          tcp
   192.168.0.1
   192.168.0.2

```

Replacing an IPv4 class entry in an existing class:

```

switch(config)# class ip MY_IP_CLASS
switch(config-class-ip)# 10 match igmp any any
switch(config-class-ip)# exit

switch(config)# do show class
Type      Name
Sequence Comment
Action    L3 Protocol
Source IP Address Source L4 Port(s)
Destination IP Address Destination L4 Port(s)
Additional Parameters
-----
IPv4      MY_IP_CLASS
10 match          igmp
   any
   any
20 ignore        udp
   any

```

```

any
30 match                                tcp
    192.168.0.1
    192.168.0.2

```

Removing an IPv4 class entry:

```

switch(config)# class ip MY_IP_CLASS
switch(config-class-ip)# no 10
switch(config-class-ip)# exit

switch(config)# do show class
Type      Name
Sequence  Comment
          Action                    L3 Protocol
          Source IP Address         Source L4 Port(s)
          Destination IP Address    Destination L4 Port(s)
          Additional Parameters
-----
IPv4      MY_IP_CLASS
20        ignore                        udp
          any
          any
30        match                        tcp
          192.168.0.1
          192.168.0.2

```

Removing an IPv4 class. Removing a class with entries removes all its entries as well. If a class associated with a policy entry (or multiple policy entries) is removed, the corresponding entries are also removed.



The corresponding entries are only removed if the class is unused by all policy entries.

```

switch(config)# no class ip MY_IP_CLASS

switch(config)# do show class
No Class found.

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config The class ip <CLASS-NAME> command takes you into the config-class-ip context where you enter the class entries.	Administrators or local user group members with execution rights for this command.

class ipv6

Syntax to create an IPv6 class and enter its context. Plus syntax to remove a class:

```
class ipv6 <CLASS-NAME>
no class ipv6 <CLASS-NAME>
```

Syntax (within the class context) for creating or removing class entries for protocols ah, gre, esp, igmp, ospf, pim (ipv6 is available as an alias for any):

```
[<SEQUENCE-NUMBER>]
{match|ignore}
{any|ipv6|ah|gre|esp|igmp|ospf|pim|<IP-PROTOCOL-NUM>}
{any|<SRC-IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]}
{any|<DST-IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count]

no <SEQUENCE-NUMBER>
```

Syntax (within the class context) for creating or removing class entries for protocols sctp, tcp, udp:

```
[<SEQUENCE-NUMBER>]
{match|ignore}
{sctp|tcp|udp}
{any|<SRC-IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>]
{any|<DST-IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>]
[urg] [ack] [psh] [rst] [syn] [fin] [established]
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count]

no <SEQUENCE-NUMBER>
```

Syntax (within the class context) for creating or removing class entries for protocol icmpv6:

```
[<SEQUENCE-NUMBER>]
{permit|deny}
{icmpv6}
{any|<SRC-IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]}
{any|<DST-IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[icmp-type {echo|echo-reply|<ICMP-TYPE-VALUE>}] [icmp-code <ICMP-CODE-VALUE>]
[dscp <DSCP-SPECIFIER>] [ecn <ECN-VALUE>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [ttl <TTL-VALUE>] [count]

no <SEQUENCE-NUMBER>
```

Syntax (within the class context) for class entry comments:

```
[<SEQUENCE-NUMBER>] comment <TEXT-STRING>

no <SEQUENCE-NUMBER> comment
```

Description

Creates or modifies an IPv6 traffic class to match specified packets. Class is composed of one or more class entries ordered and prioritized by sequence numbers. With this command, each class can classify traffic based on IPv6 header information.

The `no` form of the command deletes either an IPv6 traffic class (use `no` with the class command) or an individual IPv6 traffic class entry (use `no` with the sequence number).

Parameter	Description
ipv6	Specifies create or modify an IPv6 class.

Parameter	Description
<code><CLASS-NAME></code>	Specifies the name of this class.
<code><SEQUENCE-NUMBER></code>	Specifies a sequence number for the class entry. Optional. Range: 1-4294967295.
<code>{match ignore}</code>	Creates a rule to match or ignore specified packets.
<code><IP-PROTOCOL-NUM></code>	Specifies the protocol as its Internet Protocol number. For example, 2 corresponds to the IGMP protocol. Range: 0 to 255.
<code>{any <SRC-IP-ADDRESS>[/{<PREFIX-LENGTH> <SUBNET-MASK>}]}</code>	Specifies the source IPv4 address. <ul style="list-style-type: none"> ■ <code>any</code> - specifies any source IPv4 address. ■ <code><SRC-IP-ADDRESS></code> - specifies the source IPv4 host address. <ul style="list-style-type: none"> ○ <code><PREFIX-LENGTH></code> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 32. ○ <code><SUBNET-MASK></code> - specifies the address bits to mask (dotted decimal notation).
<code>{any <DST-IP-ADDRESS>[/{<PREFIX-LENGTH> <SUBNET-MASK>}]}</code>	Specifies the destination IPv4 address. <ul style="list-style-type: none"> ■ <code>any</code> - specifies any destination IPv4 address. ■ <code><DST-IP-ADDRESS></code> - specifies the destination IPv4 host address. <ul style="list-style-type: none"> ○ <code><PREFIX-LENGTH></code> - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 32. ○ <code><SUBNET-MASK></code> - specifies the address bits to mask (dotted decimal notation).
<code>[{eq gt lt} <PORT> range <MIN-PORT><MAX-PORT>]</code>	Specifies the port or port range. Port numbers are in the range of 0 to 65535. <ul style="list-style-type: none"> ■ <code>eq <PORT></code> - specifies the Layer 4 port. ■ <code>gt <PORT></code> - specifies any Layer 4 port greater than the indicated port. ■ <code>lt <PORT></code> - specifies any Layer 4 port less than the indicated port.

Parameter	Description
	<ul style="list-style-type: none"> ■ <code>range <MIN-PORT> <MAX-PORT></code> - specifies the Layer 4 port range.
<code>urg, ack, psh, rst, syn, fin, established</code>	These TCP flag matching parameters are not supported.
<code>dscp <DSCP-SPECIFIER></code>	Specifies the Differentiated Services Code Point (DSCP), either a numeric <code><DSCP-VALUE></code> (0 to 63) or one of these keywords: <ul style="list-style-type: none"> ■ <code>AF11</code> - DSCP 10 (Assured Forwarding Class 1, low drop probability) ■ <code>AF12</code> - DSCP 12 (Assured Forwarding Class 1, medium drop probability) ■ <code>AF13</code> - DSCP 14 (Assured Forwarding Class 1, high drop probability) ■ <code>AF21</code> - DSCP 18 (Assured Forwarding Class 2, low drop probability) ■ <code>AF22</code> - DSCP 20 (Assured Forwarding Class 2, medium drop probability) ■ <code>AF23</code> - DSCP 22 (Assured Forwarding Class 2, high drop probability) ■ <code>AF31</code> - DSCP 26 (Assured Forwarding Class 3, low drop probability) ■ <code>AF32</code> - DSCP 28 (Assured Forwarding Class 3, medium drop probability) ■ <code>AF33</code> - DSCP 30 (Assured Forwarding Class 3, high drop probability) ■ <code>AF41</code> - DSCP 34 (Assured Forwarding Class 4, low drop probability) ■ <code>AF42</code> - DSCP 36 (Assured Forwarding Class 4, medium drop probability) ■ <code>AF43</code> - DSCP 38 (Assured Forwarding Class 4, high drop probability) ■ <code>CS0</code> - DSCP 0 (Class Selector 0: Default) ■ <code>CS1</code> - DSCP 8 (Class Selector 1:

Parameter	Description
	Scavenger) <ul style="list-style-type: none"> ■ CS2 - DSCP 16 (Class Selector 2: OAM) ■ CS3 - DSCP 24 (Class Selector 3: Signaling) ■ CS4 - DSCP 32 (Class Selector 4: Real time) ■ CS5 - DSCP 40 (Class Selector 5: Broadcast video) ■ CS6 - DSCP 48 (Class Selector 6: Network control) ■ CS7 - DSCP 56 (Class Selector 7) ■ EF - DSCP 46 (Expedited Forwarding)
ecn <ECN-VALUE>	Specifies an Explicit Congestion Notification value. Range: 0 to 3.
ip-precedence <IP-PRECEDENCE-VALUE>	Specifies an IP precedence value. Range: 0 to 7.
tos <TOS-VALUE>	Specifies the Type of Service value. Range: 0 to 31.
fragment	Specifies a fragment packet.
vlan <VLAN-ID>	Specifies VLAN tag to match on. 802.1Q VLAN ID. NOTE: This parameter cannot be used in any class that will be applied to a VLAN.
ttl <TTL-VALUE>	Specifies a time-to-live (hop limit) value. Range: 0 to 255.
count	Keeps the hit counts of the number of packets matching this class entry.
[<SEQUENCE-NUMBER>] comment <TEXT-STRING>	Adds a comment to a class entry. The <code>no</code> form removes only the comment from the class entry.

Usage

- If you enter an existing <CLASS-NAME> value, the existing class is modified with any new <SEQUENCE-NUMBER> value. This action creates an additional class entry. Any existing <SEQUENCE-NUMBER> value replaces the existing class entry with the same sequence number.
- If no sequence number is specified, a new class entry is appended to the end of the class with a sequence number equal to the highest class entry currently in the list plus 10.
- If the <IP-PROTOCOL-NUM> parameter is used instead of a protocol name, ensure that any needed class entry-definition parameters specific to the selected protocol are also provided.

Examples

Creating an IPv6 class with two entries:

```
switch(config)# class ipv6 MY_IPV6_CLASS
switch(config-class-ipv6)# 10 match icmpv6 any any
switch(config-class-ipv6)# 20 ignore udp any any
switch(config-class-ipv6)# exit

switch(config)# do show class
Type      Name
Sequence Comment
          Action                L3 Protocol
          Source IP Address      Source L4 Port(s)
          Destination IP Address  Destination L4 Port(s)
          Additional Parameters
-----
IPv6      MY_IPV6_CLASS
          10 match                    icmpv6
          any
          any
          20 ignore                    udp
          any
          any
```

Adding a comment to an existing IPv6 class entry:

```
switch(config)# class ipv6 MY_IPV6_CLASS
switch(config-class-ipv6)# 10 match icmpv6 any any
switch(config-class-ipv6)# 20 ignore udp any any
switch(config-class-ipv6)# 20 comment myipv6class
switch(config-class-ipv6)# exit

switch(config)# do show class
Type      Name
Sequence Comment
          Action                L3 Protocol
          Source IP Address      Source L4 Port(s)
          Destination IP Address  Destination L4 Port(s)
          Additional Parameters
-----
IPv6      MY_IPV6_CLASS
          10 match                    icmpv6
          any
          any
          20 myipv6class
          ignore                    udp
          any
          any
```

Removing a comment from an existing IPv6 class entry:

```
switch(config)# class ipv6 MY_IPV6_CLASS
switch(config-class-ipv6)# no 20 comment
switch(config-class-ipv6)# exit

switch(config)# do show class
Type      Name
```

```

Sequence Comment
Action
Source IP Address
Destination IP Address
Additional Parameters
-----
IPv6      MY_IPV6_CLASS
10 match
   any
   any
20 ignore
   any
   any
L3 Protocol
Source L4 Port(s)
Destination L4 Port(s)

```

Replacing an IPv6 class entry in an existing IPv6 class:

```

switch(config)# class ipv6 MY_IPV6_CLASS
switch(config-class-ipv6)# 10 match any any 1020::
switch(config-class-ipv6)# exit

switch(config)# do show class
Type      Name
Sequence Comment
Action
Source IP Address
Destination IP Address
Additional Parameters
-----
IPv6      MY_IPV6_CLASS
10 match
   any
   1020::
20 ignore
   any
   any
L3 Protocol
Source L4 Port(s)
Destination L4 Port(s)

```

Removing an IPv6 class entry:

```

switch(config)# class ipv6 MY_IPV6_CLASS
switch(config-class-ipv6)# no 10
switch(config-class-ipv6)# exit

switch(config)# do show class
Type      Name
Sequence Comment
Action
Source IP Address
Destination IP Address
Additional Parameters
-----
IPv6      MY_IPV6_CLASS
20 ignore
   any
   any
L3 Protocol
Source L4 Port(s)
Destination L4 Port(s)

```

Removing an IPv6 class. Removing a class with entries removes all its entries as well. If a class associated with a policy entry (or multiple policy entries) is removed, the corresponding entries are also removed.



The corresponding entries are only removed if the class is unused by all policy entries.

```
switch(config)# no class ipv6 MY_IPV6_CLASS

switch(config)# do show class
No Class found.
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	<p>config</p> <p>The class ipv6 <CLASS-NAME> command takes you into the config-class-ipv6 command context where you enter the class entries.</p>	Administrators or local user group members with execution rights for this command.

class mac

class mac <CLASS-NAME>

```
[<SEQUENCE-NUMBER>]
{match|ignore}
{any|<SRC-MAC-ADDRESS>[/<ETHERNET-MASK>]}
{any|<DST-MAC-ADDRESS>[/<ETHERNET-MASK>]}
{any|aarp|appletalk|arp|fcoe|fcoe-init|ip|ipv6|ipx-arp|ipx-non-arp|is-is|
  lldp|mpls-multicast|mpls-unicast|q-in-q|rbridge|trill|wake-on-lan|
  <NUMERIC-ETHERTYPE>}
[pcp <PCP-VALUE>] [vlan <VLAN-ID>] [count]

[<SEQUENCE-NUMBER>] comment <TEXT-STRING>
```

Description

Creates or modifies a MAC traffic class to match specified packets. Class is composed of one or more class entries ordered and prioritized by sequence numbers. With this command, each class can classify traffic based on MAC header information.

The `no` form of the command can be used to delete either a MAC traffic class (use `no` with the class command) or an individual MAC traffic class entry (use `no` with the sequence number).

Parameter	Description
mac	Specifies create or modify a MAC class.

Parameter	Description
<CLASS-NAME>	Specifies the name of this class.
<SEQUENCE-NUMBER>	Specifies a sequence number for the class entry. Optional. Range: 1-4294967295.
{match ignore}	Creates a rule to match or ignore specified packets.
comment	Stores the remaining entered text as a class comment.
{any <SRC-MAC-ADDRESS>[/<ETHERNET-MASK>]}	Specifies the source host MAC address (xxxx.xxxx.xxxx), OUI, or the keyword <code>any</code> . You can optionally include the following: <ETHERNET-MASK> - The address bits to mask (xxxx.xxxx.xxxx).
{any <DST-MAC-ADDRESS>[/<ETHERNET-MASK>]}	Specifies the destination host MAC address (xxxx.xxxx.xxxx), OUI, or the keyword <code>any</code> . You can optionally include the following: <ETHERNET-MASK> - The address bits to mask (xxxx.xxxx.xxxx).
Protocol	Select an ethertype protocol from the following (enter one only): <ul style="list-style-type: none"> ■ <code>any</code> - Any ethertype protocol ■ <NUMERIC-ETHERTYPE> - Enter an EtherType protocol number. Range: 0x600-0xffff. ■ Or enter an EtherType protocol name from the following list: <ul style="list-style-type: none"> ○ <code>aarp</code> ○ <code>appletalk</code> ○ <code>arp</code> ○ <code>fcoe</code> ○ <code>fcoe-init</code> ○ <code>ip</code> ○ <code>ipv6</code> ○ <code>ipx-arpa</code> ○ <code>ipx-non-arpa</code> ○ <code>is-is</code> ○ <code>lldp</code> ○ <code>mpls-multicast</code> ○ <code>mpls-unicast</code> ○ <code>q-in-q</code> ○ <code>rbridge</code> ○ <code>trill</code> ○ <code>wake-on-lan</code>

Parameter	Description
pcp <PCP-VALUE>	Specifies matching on QoS Priority Code Point. Range: 0-7.
vlan <VLAN-ID>	Specifies matching on a VLAN ID. Enter a VLAN ID or the VLAN name, if configured. NOTE: This parameter cannot be used in any class that will be applied to a VLAN.
count	Keeps the hit counts of the number of packets matching this class entry.

Examples

Creating a MAC class:

```
switch(config)# class mac MY_MAC_CLASS
switch(config-class-mac)# match any any lldp
switch(config-class-mac)# ignore any any arp
switch(config-class-mac)# exit
switch(config)# do show class
Type      Name
Sequence Comment          Action          EtherType
Source MAC Address
Destination MAC Address
Additional Parameters
-----
MAC      MY_MAC_CLASS
10 match any any          lldp
20 ignore any any          arp
```

Adding a comment to an existing MAC class entry:

```
switch(config)# class mac MY_MAC_CLASS
switch(config-class-mac)# 10 comment MY_CLASS_ENTRY
switch(config-class-mac)# exit
switch(config)# do show class
Type      Name
Sequence Comment          Action          EtherType
Source MAC Address
Destination MAC Address
Additional Parameters
-----
MAC      MY_MAC_CLASS
10 MY_CLASS_ENTRY match any any          lldp
20 ignore any any          arp
```



```
any
any
```

Removing a comment from an existing MAC class entry:

```
switch(config)# class mac MY_MAC_CLASS
switch(config-class-mac)# no 10 comment MY_CLASS_ENTRY
switch(config-class-mac)# exit
switch(config)# do show class
Type      Name
Sequence  Comment
          Action                    EtherType
          Source MAC Address
          Destination MAC Address
          Additional Parameters
-----
MAC       MY_MAC_CLASS
10 match
   any
   any
20 ignore
   any
   any
```

Replacing a MAC class entry in an existing MAC class:

```
switch(config)# class mac MY_MAC_CLASS
switch(config-class-mac)# 10 match any any any
switch(config-class-mac)# exit
switch(config)# do show class
Type      Name
Sequence  Comment
          Action                    EtherType
          Source MAC Address
          Destination MAC Address
          Additional Parameters
-----
MAC       MY_MAC_CLASS
10 match
   any
   any
20 ignore
   any
   any
```

Removing a MAC class entry:

```
switch(config)# class mac MY_MAC_CLASS
switch(config-class-mac)# no 1
switch(config-class-mac)# exit
switch(config)# do show class
Type      Name
Sequence  Comment
          Action                    EtherType
          Source MAC Address
          Destination MAC Address
          Additional Parameters
```

```

-----
MAC          MY_MAC_CLASS
              2 ignore          arp
              any
              any

```

Removing a MAC class. Removing a class with entries removes all its entries as well. If a class associated with a policy entry (or multiple policy entries) is removed, the corresponding entries are also removed.



The corresponding entries are only removed if the class is unused by all policy entries.

```

switch(config)# no class mac MY_MAC_CLASS
switch(config)# do show class
No Class found.

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
8400	config The class mac <CLASS-NAME> command takes you into the config-class-mac context where you enter the class entries.	Administrators or local user group members with execution rights for this command.

class resequence

```
class {ip|ipv6|mac} <CLASS-NAME> resequence <STARTING-SEQUENCE-NUMBER> <INCREMENT>
```

Description

Resequencing numbering in an IPv4, IPv6, or MAC class.

Parameter	Description
{ip ipv6 mac} <CLASS-NAME>	Specifies the class where you want to resequence class entries.
<STARTING-SEQUENCE-NUMBER>	Specifies the sequence number to start resequencing from.
<INCREMENT>	Specifies how much to increment the sequence numbers by.

Examples

Resequencing an IPv4 class:

```

switch(config)# class ip MY_IP_CLASS resequence 1 10
switch(config)# do show class
Type          Name
  Sequence    Comment
            Action          L3 Protocol
            Source IP Address Source L4 Port(s)
            Destination IP Address Destination L4 Port(s)
            Additional Parameters
-----
IPv4          MY_IP_CLASS
  1 match          igmp
    any
    any
  11 ignore        udp
    any
    any
  21 match          tcp
    192.168.0.1
    192.168.0.2

```

Resequencing an IPv6 class:

```

switch(config)# class ipv6 MY_IPV6_CLASS resequence 1 1
switch(config-class-ipv6)# exit
switch(config)# do show class
Type          Name
  Sequence    Comment
            Action          L3 Protocol
            Source IP Address Source L4 Port(s)
            Destination IP Address Destination L4 Port(s)
            Additional Parameters
-----
IPv6          MY_IPV6_CLASS
  1 match          any
    any
    1020::
  2 ignore        udp
    any
    any

```

Resequencing a MAC class:

```

switch(config)# class mac MY_MAC_CLASS resequence 1 1
switch(config)# do show class
Type          Name
  Sequence    Comment
            Action          EtherType
            Source MAC Address
            Destination MAC Address
            Additional Parameters
-----
MAC          MY_MAC_CLASS
  1 match          any
    any
    any
  2 ignore        arp
    any
    any

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

class reset

```
class { all | ip <CLASS-NAME> | ipv6 <CLASS-NAME> | mac <CLASS-NAME> } reset
```

Description

Changes the user-specified class configuration to match the active class configuration. Use this command when there is a discrepancy between what the user configured and what is active and accepted by the system.

Parameter	Description
{ all ip <CLASS-NAME> ipv6 <CLASS-NAME> mac <CLASS-NAME> }	Specifies either all classes be reset or specifies the type (ip for IPv4, ipv6 for IPv6 or mac for MAC ACL) and name of the class to be reset.

Examples

Resetting the user-specified configuration to the active configuration:

```
switch(config)# class all reset
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

clear policy hitcounts

```
clear policy hitcounts { all | [<POLICY-NAME>] [[interface <IF-NAME> [in|routed-in]] | [vlan <VLAN-ID> [in]]] | global }
```

Description

Clears the policy hit count statistics.

Parameter	Description
all	Selects all policies.
<POLICY-NAME>	Specifies the policy name.
interface <IF-NAME>	Specifies the interface name.
vlan <VLAN-ID>	Specifies the VLAN.
in	Specifies the inbound (ingress) traffic direction.
routed-in	Selects the routed in traffic direction. Not applicable to a policy applied to a VLAN.
global	Selects the globally applied policy.

Examples

Clearing policy hit counts and then showing the policy hit counts (statistics):

```
switch# clear policy hitcounts my_policy int 1/1/1 in
switch# show policy hitcounts my_policy
Statistics for Policy my_policy:
Interface 1/1/1* (in):
    Hit Count  Configuration
10 class ipv6 my_class1 action dscp af21 action drop
    0 10 match any any any count
* policy statistics are shared among each context type (interface, VLAN).
  For routed ingress, they are only shared within the same VRF.
  Use 'policy NAME copy' to create a new policy for separate statistics.
```

Clearing the globally applied policy hit counts and then showing the global policy hit counts (statistics):

```
switch# clear policy hitcounts global
switch# show policy hitcounts global
Statistics for Policy global:
Global Policy:
    Hit Count  Configuration
10 class ipv6 my_class1 action mirror
    0 10 match any any any count
* policy statistics are shared among each context type (interface, VLAN).
  For routed ingress, they are only shared within the same VRF.
  Use 'policy NAME copy' to create a new policy for separate statistics.
```

Clearing hit counts for policy MY_IPv6_Policy applied to VLAN 10 (ingress):

```
switch# clear policy hitcounts My_IPv6_Policy vlan 10 in
```

Clearing hit counts for all policies:

```
switch# clear policy hitcounts all
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

policy

policy <POLICY-NAME>

```
[<SEQUENCE-NUMBER>]
class {ip|ipv6|mac} <CLASS-NAME>
    action {<REMARK-ACTIONS> | <POLICE-ACTIONS> | <OTHER-ACTIONS>}
    [{<REMARK-ACTIONS> | <POLICE-ACTIONS> | <OTHER-ACTIONS>}]

[<SEQUENCE-NUMBER>]
comment ...
```

Description

Creates or modifies classifier policy and policy entries. A policy is made up of one or more policy entries ordered and prioritized by sequence numbers. Each entry has an IPv4/IPv6/MAC class and zero or more policy actions associated with it.

A policy must be applied using the `apply` command.

The `no` form of the command can be used to delete either a policy (use `no` with the policy command) or an individual policy entry (use `no` with the sequence number).

Parameter	Description
<POLICY-NAME>	Specifies the name of the policy.
<SEQUENCE-NUMBER>	Specifies a sequence number for the policy entry. Optional. Range: 1 to 4294967295.
comment	Stores the remaining entered text as a policy entry comment.
class {ip ipv6 mac} <CLASS-NAME>	Specifies a type of class, <code>ip</code> for IPv4, <code>ipv6</code> for IPv6 and <code>mac</code> for a MAC policy. And specifies a class name.
<REMARK-ACTIONS>	Remark actions can be any of the following options: { <code>pcp <PRIORITY></code> <code>ip-precedence <IP-PRECEDENCE_VALUE></code> <code>dscp <DSCP-VALUE></code> <code>local-priority <LOCAL-PRIORITY-VALUE></code> } where:
<code>pcp <PCP-VALUE></code>	Specifies the Priority Code Point (PCP) value. Range: 0 to 7.

Parameter	Description
<code>ip-precedence <IP-PRECEDENCE-VALUE></code>	Specifies the numeric IP precedence value. Range: 0 to 7.
<code>dscp <DSCP-VALUE></code>	<p>Specifies a Differentiated Services Code Point (DSCP) value. Enter either a numeric value (0 to 63) or a keyword as follows:</p> <ul style="list-style-type: none"> ■ AF11 - DSCP 10 (Assured Forwarding Class 1, low drop probability) ■ AF12 - DSCP 12 (Assured Forwarding Class 1, medium drop probability) ■ AF13 - DSCP 14 (Assured Forwarding Class 1, high drop probability) ■ AF21 - DSCP 18 (Assured Forwarding Class 2, low drop probability) ■ AF22 - DSCP 20 (Assured Forwarding Class 2, medium drop probability) ■ AF23 - DSCP 22 (Assured Forwarding Class 2, high drop probability) ■ AF31 - DSCP 26 (Assured Forwarding Class 3, low drop probability) ■ AF32 - DSCP 28 (Assured Forwarding Class 3, medium drop probability) ■ AF33 - DSCP 30 (Assured Forwarding Class 3, high drop probability) ■ AF41 - DSCP 34 (Assured Forwarding Class 4, low drop probability) ■ AF42 - DSCP 36 (Assured Forwarding Class 4, medium drop probability) ■ AF43 - DSCP 38 (Assured Forwarding Class 4, high drop probability) ■ CS0 - DSCP 0 (Class Selector 0: Default) ■ CS1 - DSCP 8 (Class Selector 1: Scavenger) ■ CS2 - DSCP 16 (Class Selector 2: OAM) ■ CS3 - DSCP 24 (Class Selector 3: Signaling) ■ CS4 - DSCP 32 (Class Selector 4: Real time) ■ CS5 - DSCP 40 (Class Selector 5: Broadcast video) ■ CS6 - DSCP 48 (Class Selector 6: Network control) ■ CS7 - DSCP 56 (Class Selector 7) ■ EF - DSCP 46 (Expedited Forwarding)
<code>local-priority <LOCAL-PRIORITY-VALUE></code>	Specifies a local priority value. Range: 0 to 7.
<code><POLICE-ACTIONS></code>	Police actions can be the following {cir <RATE-BPS>cbs <BYTES> exceed} where:
<code>cir kbps <RATE-KBPS></code>	Specifies a Committed Information Rate value in Kilobits per second. Range: 1 to 4294967295.
<code>cbs <BYTES></code>	Specifies a Committed Burst Size value in bytes. Range: 1 to 4294967295.

Parameter	Description
exceed	Specifies action to take on packets that exceed the rate limit.
<OTHER-ACTIONS>	Other actions can be the following:
drop	Specifies drop traffic.

Usage

- An applied policy will process a packet sequentially against policy entries in the list until the last policy entry in the list has been evaluated or the packet matches an entry.
- Entering an existing <POLICY-NAME> value will cause the existing policy to be modified, with any new <SEQUENCE-NUMBER> value creating an additional policy entry, and any existing <SEQUENCE-NUMBER> value replacing the existing policy entry with the same sequence number.
- If no sequence number is specified, a new policy entry will be appended to the end of the entry list with a sequence number equal to the highest policy entry currently in the list plus 10.
- On an 8400 switch, IP remarking (ip-precedence, dscp) will only be performed if the packet is routed.

Examples

Creating a policy with several entries:

```
switch(config)# policy MY_POLICY
switch(config-policy)# 10 class ipv6 MY_CLASS1 action dscp af21 action drop
switch(config-policy)# 20 class ip MY_CLASS3 action mirror 1
switch(config-policy)# exit
switch(config)# do show policy
      Name
      Sequence Comment
      Class Type
      action
-----
      MY_POLICY
      10
      MY_CLASS1 ipv6
      drop
      dscp AF21

      20
      MY_CLASS3 ipv4
      mirror 1
```

Adding a comment to an existing policy entry:

```
switch(config)# policy MY_POLICY
switch(config-policy)# 20 comment MY_TEST_POLICY
switch(config-policy)# exit
switch(config)# do show policy
      Name
      Sequence Comment
      Class Type
      action
-----
      MY_POLICY
```



```

10
  MY_CLASS1 ipv6
    drop
    dscp AF21

20 MY_TEST_POLICY
  MY_CLASS3 ipv4
    mirror 1

```

Removing a comment from an existing policy entry:

```

switch(config)# policy MY_POLICY
switch(config-policy)# no 20 comment
switch(config-policy)# exit
switch(config)# do show policy
  Name
  Sequence Comment
  Class Type
  action
-----
  MY_POLICY
10
  MY_CLASS1 ipv6
    drop
    dscp AF21

20
  MY_CLASS3 ipv4
    mirror 1

```

Adding/Replacing a policy entry in an existing policy:

```

switch(config)# policy MY_POLICY
switch(config-policy)# 10 class ip MY_CLASS3 action drop action dscp af21
switch(config-policy)# exit
switch(config)# do show policy
  Name
  Sequence Comment
  Class Type
  action
-----
  MY_POLICY
10
  MY_CLASS3 ipv4
    drop
    dscp AF21

20
  MY_CLASS3 ipv4
    mirror 1

```

Removing a policy entry:

```

switch(config)# policy MY_POLICY
switch(config-policy)# no 10
switch(config-policy)# exit
switch(config)# do show policy

```

```

Name
Sequence Comment
Class Type
          action
-----
MY_POLICY
20
MY_CLASS3 ipv4
          mirror 1

```

Removing a policy:

```

switch(config)# no policy MY_POLICY
switch(config)# do show policy
Name
Sequence Comment
Class Type
          action
-----
MY_POLICY2
2
MY_CLASS3 ipv4
          mirror 1

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config The <code>policy</code> command takes you into the <code>config-policy</code> context where you enter the policy entries.	Administrators or local user group members with execution rights for this command.

policy copy

`policy <POLICY-NAME> copy <DESTINATION-POLICY>`

Description

Copies a policy to a new destination policy or overwrites an existing policy. Copying a policy copies all its entries as well.

Parameter	Description
<POLICY-NAME>	Specifies the policy to be copied.

Parameter	Description
<DESTINATION-POLICY>	Specifies the name of the destination policy.

Examples

Copying a policy:

```
switch(config)# policy MY_POLICY copy MY_POLICY2
switch(config)# do show policy
      Name
      Sequence Comment
      Class Type
      action
-----
      MY_POLICY
      2
      my_class3 ipv4
      mirror 1
-----
      MY_POLICY2
      2
      my_class3 ipv4
      mirror 1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

policy resequence

policy <POLICY-NAME> resequence <STARTING-SEQ-NUM> <INCREMENT>

Description

Resequences numbering in a policy.

Parameter	Description
<POLICY-NAME>	Specifies the policy where you want to resequence policy entries.
<STARTING-SEQ-NUM>	Specifies the sequence number to start resequencing from.
<INCREMENT>	Specifies how much to increment the sequence numbers by.

Examples

Resequencing a policy:

```
switch(config)# policy MY_POLICY resequence 1 1
switch(config)# do show policy
      Name
      Sequence Comment
      Class Type
      action
-----
      MY_POLICY
      1
      MY_CLASS3 ipv4
      drop
      dscp AF21
      2
      MY_CLASS3 ipv4
      mirror 1
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

policy reset

policy <POLICY-NAME> reset

Description

Changes the user-specified policy configuration to match the active policy configuration. Use this command when a discrepancy exists between what the user configured and what is active and accepted by the system.

Parameter	Description
<POLICY-NAME>	Specifies the policy to be reset.

Examples

Resetting a policy:

```
switch(config)# policy MY_POLICY reset
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	config	Administrators or local user group members with execution rights for this command.

show class

```
show class [ip | ipv6 | mac] [<CLASS-NAME>] [commands] [configuration]
```

Description

Shows class configuration information.

All parameters are optional.

Parameter	Description
[ip ipv6 mac]	Selects the class type for the display: <code>ip</code> for IPv4, <code>ipv6</code> for IPv6, or <code>mac</code> for MAC classes.
<CLASS-NAME>	Specifies the class name.
commands	Specifies whether to display output as the CLI commands showing the configured class entries.
configuration	Specifies whether to display classes that have been configured by the user, even if they are not active due to issues with the command parameters or hardware issues. This parameter is useful during a mismatch between the entered configuration and the previous successfully programmed (active) classes.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing all class configuration:

```
switch# show class
Type Name
  Sequence Comment
    action          L3 Protocol
    Source IP address Source L4 Port(s)
    Destination IP address Destination L4 Port(s)
    Additional Parameters
-----
ipv4 MY_IPV4_CLASS
  10 my first class entry comment
    match          icmp
```

```

192.168.0.1/255.255.255.0
192.168.1.1/255.255.255.0
VLAN: 1
20 my second class entry comment
   ignore                tcp
10.100.0.10/255.255.255.0 < 3000
10.100.1.10/255.255.255.0 > 2000
VLAN: 1
-----

```

Showing class configuration for the IPv4 class MY_IPV4_CLASS as CLI commands:

```

switch# show class ip MY_IPV4_CLASS commands
class ip "MY_IPV4_CLASS"
  10 match icmp 192.168.0.1/255.255.255.0 192.168.1.1/255.255.255.0 vlan 1
  10 comment my first class entry comment
  20 ignore tcp 10.100.0.10/255.255.255.0 lt 3000 10.100.1.10/255.255.255.0 gt
    2000 vlan 1
  20 comment my second class entry comment

```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

show policy

Syntax that shows information for all policies:

```
show policy [commands] [configuration] [vsx-peer]
```

Syntax that filters by policies applied to an interface or VLAN:

```
show policy [interface <IF-NAME> [in | routed-in] | vlan <VLAN-ID> [in]]
           [commands] [configuration] [vsx-peer]
```

Syntax that filters by the named policy:

```
show policy <POLICY-NAME> [commands] [configuration] [vsx-peer]
```

Syntax that filters by the globally applied policy:

```
show policy global [commands] [configuration] [vsx-peer]
```

Syntax that shows statistical information in the form of hit counts:

```
show policy hitcounts <POLICY-NAME> [interface <IF-NAME> [in | routed-in] |
                                   vlan <VLAN-ID> [in]] [vsx-peer]
```

Syntax that shows statistical information in the form of hit counts for the globally applied policy:

```
show policy hitcounts global [vsx-peer]
```

Description

Shows information about your defined policies and where they have been applied. When `show policy` is entered without parameters, information for all policies is shown. The parameters filter the list of policies for which information is shown.

Available filtering includes:

- The content of a specific policy.
- All policies applied to a specific interface.
- All policies applied to a specific VLAN.
- The globally applied policy.

To display policy statistics, use the `show policy hitcounts` form of this command.



When a policy is applied to a physical interface or lag using command `apply policy`, with the `per-interface` parameter included, unique instances of the policy are applied to each physical interface port or LAG. The unique instance of a policy has a parent-child relationship with the policy from which it was created. The `show policy` command shows information about the parent policy not the unique instances.



If a policy contains any class entries with the `count` keyword and policy entries with the `cir` action, and the policy is applied to multiple physical or virtual interfaces in the same direction, except for the routed ingress direction, the statistics will be aggregated. In the routed ingress direction, the statistics will be aggregated in multiple physical or virtual interfaces in the same VRF. If separate statistics for different physical or virtual interfaces are required, then another policy should be created. Alternatively, in the case of physical interfaces or LAGs, a policy applied with `per-interface set` can be used.

Parameter	Description
<code>interface <IF-NAME></code>	Specifies the interface name.
<code>vlan <VLAN-ID></code>	Specifies the VLAN.
<code>in</code>	Selects the inbound (ingress) traffic direction.
<code>routed-in</code>	Selects the routed in traffic direction. Not applicable to a policy applied to a VLAN.
<code><POLICY-NAME></code>	Specifies the policy name.
<code>commands</code>	Causes the policy definition to be shown as the commands and parameters used to create it rather than in tabular form.
<code>configuration</code>	Causes the user-configured policies be shown as entered, even if the policies are not active due to policy-definition command issues or hardware issues. This parameter is useful if there is a mismatch between the entered configuration and the previous successfully programmed (active) policies configuration.
<code>global</code>	Selects the globally applied policy.
<code>hitcounts</code>	Selects the policy hit counts (statistics).
<code>vsx-peer</code>	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing information for all policies:

```
switch# show policy
      Name
Sequence Comment
      Class Type
              action
-----
      my_policy
10 QOS class
   class1 ipv4
       dscp af21
       drop
20 PBR policy.
   class2 ipv4
       pbr mypbr
-----
```

Showing a policy as commands:

```
switch# show policy commands
policy my_policy
10 class ip class1 action dscp af21 action drop
20 class ip class2 action pbr mypbr
```

Showing the globally applied policy:

```
switch# show policy global commands
policy global1
10 class ip my_class1 action drop
apply policy my_policy in
```

Showing policy hit counts (statistics) for the globally applied policy: :

```
switch# show policy hitcounts global
Statistics for Policy My_Policy:
global (in):
  Matched Packets Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    0 10 match tcp any any count [ 0 kbps conform ]
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]
```

Showing policy hit counts (statistics) for a policy applied everywhere (with 1/1/1 and 1/1/5 being applied per interface):

```
switch# show policy hitcounts My_Policy
Statistics for Policy My_Policy:

Interface 1/1/1,lag1 (in):
```



```

    Matched Packets Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    - 10 match tcp any any ack
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]

Interface 1/1/4 (in):
    Matched Packets Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    0 10 match tcp any any count [ 0 kbps conform ]
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]

Interface 1/1/5 (in):
    Matched Packets Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    0 10 match tcp any any count [ 0 kbps conform ]
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]

interface 1/1/2.10,1/1/3.10 (in):
    Matched Packets Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    - 10 match tcp any any ack
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]
...

```

Showing policy hit counts (statistics) for a policy applied on physical interfaces and LAGs:

```

switch# show policy hitcounts My_Policy interface 1/1/1
Statistics for Policy My_Policy:

Interface 1/1/1,lag1 (in):
    Matched Packets Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    0 10 match tcp any any count [ 0 kbps conform ]
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]

```

Showing policy hit counts (statistics) for a policy applied on VLANs:

```

switch# show policy hitcounts My_Policy vlan 30
Statistics for Policy My_Policy:

vlan 30 (in):
    Matched Packets  Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    0 10 match tcp any any count [ 0 kbps conform ]
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]

```

Showing policy hit counts (statistics) for a policy applied on interface VLANs:

```

switch# show policy hitcounts My_Policy interface vlan10
Statistics for Policy My_Policy:

VRF red
interface vlan 10 (routed-in):
    Matched Packets  Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    0 10 match tcp any any count [ 0 kbps conform ]
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]

```

Showing policy hit counts (statistics) for a policy applied on interface VLANs for a specific VRF:

```

switch# show policy hitcounts My_Policy vrf red routed-in
Statistics for Policy My_Policy:

VRF red
interface vlan 10 (routed-in):
    Matched Packets  Configuration
10 class ip My_ip_Class
    0 10 match tcp any any ack count
    - 20 match udp any lt 8 any
    0 30 match icmp any 10.1.1.10 count
20 class ipv6 My_ipv6_Class action cir kbps 1000000 cbs 1000000 exceed drop
    0 10 match tcp any any count [ 0 kbps conform ]
    0 20 match icmpv6 1000::10 any count [ 0 kbps conform ]

```

Command History

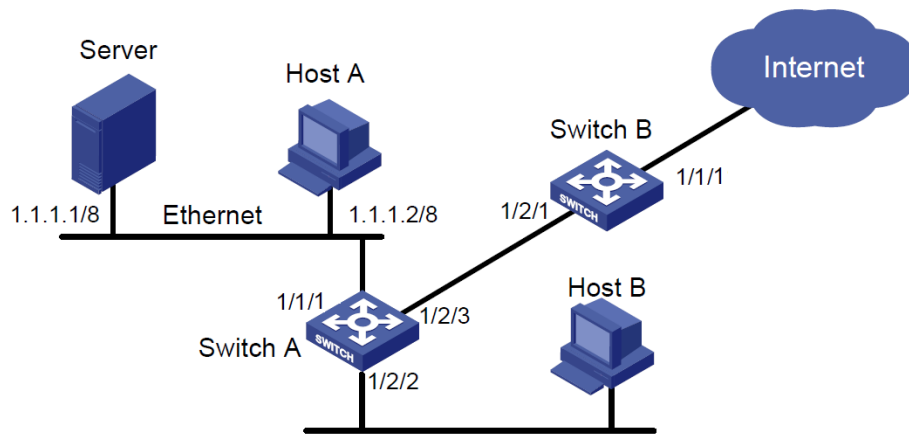
Release	Modification
10.08	Added [per-interface] information. Updated examples.
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
All platforms	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

This example configures traffic policing on:

- A 10-Gbit Ethernet of Switch A meeting the following requirements:
 - Police the rate of packets from the server to 102,400 kbps. Traffic 102,400 kbps or less is forwarded. The traffic more than 102,400 kbps is dropped.
 - Police the rate of packets from Host A to 25,600 kbps. Traffic 25,600 kbps or less is forwarded. The traffic more than 25,600 kbps is dropped.
- A 10-Gbit Ethernet 1/2/1 of Switch B limiting the incoming traffic rate of HTTP packets on 10-Gbit Ethernet 1/1/1 to the data rate of 204,800 kbps and dropping excess packets.



Configuring the classifier policies example

These steps are part of the classifier policies configuration example.

Procedure

1. Configure Switch A.

Create traffic classes named SERVER_TRAFFIC and HOST_A_TRAFFIC for matching the packets from the server and Host A:

```
switch# configure
switch(config)# class ip SERVER_TRAFFIC
switch(config-class-ip)# match any 1.1.1.1 any
switch(config-class-ip)# exit
switch(config)# class ip HOST_A_TRAFFIC
switch(config-class-ip)# match any 1.1.1.2 any
switch(config-class-ip)# exit
```

2. Create a classifier policy named RATE_LIMIT_POLICY:

```
switch(config)# policy RATE_LIMIT_POLICY
```

3. Configure the policy RATE_LIMIT_POLICY, so that 102,400 kbps of traffic, matching the class SERVER_TRAFFIC, is forwarded and the excess is dropped:

```
switch(config-policy)# class ip SERVER_TRAFFIC action cir kbps 102400 exceed drop
```

4. Configure the policy RATE_LIMIT_POLICY so that 25,600 kbps of traffic, matching the class HOST_A_TRAFFIC, is forwarded and the excess is dropped:

```
switch(config-policy)# class ip HOST_A_TRAFFIC action cir kbps 25600 exceed drop  
switch(config-policy)# exit
```

5. Apply RATE_LIMIT_POLICY to interface 1/1/1 for the inbound traffic:

```
switch(config)# int 1/1/1  
switch(config-if)# apply policy RATE_LIMIT_POLICY in  
switch(config-if)# exit
```

6. To view the configuration with the RATE_LIMIT_POLICY applied:

```
switch# show running-config  
Current configuration:  
!  
...  
class ip SERVER_TRAFFIC  
  10 match any 1.1.1.1 any  
class ip HOST_A_TRAFFIC  
  10 match any 1.1.1.2 any  
policy RATE_LIMIT_POLICY  
  10 class ip SERVER_TRAFFIC action cir kbps 102400 exceed drop  
  20 class ip HOST_A_TRAFFIC action cir kbps 25600 exceed drop  
interface 1/1/1  
  apply policy RATE_LIMIT_POLICY in
```

7. Configure Switch B.

Create a traffic class named HTTP_TRAFFIC and configure it to match traffic to port 80:

```
switch(config)# class ip HTTP_TRAFFIC  
switch(config-class-ip)# match tcp any any eq 80  
switch(config-class-ip)# exit
```

8. Create a classifier policy named RATE_LIMIT_HTTP:

```
switch(config)# policy RATE_LIMIT_HTTP
```

- Configure the policy RATE_LIMIT_HTTP so that 204,800 kbps of traffic, matching the class HTTP_TRAFFIC, is forwarded and the excess is dropped:

```
switch(config-policy)# class ip HTTP_TRAFFIC action cir kbps 204800 exceed drop
switch(config-policy)# exit
```

- Apply RATE_LIMIT_HTTP to interface 1/1/1 for inbound traffic:

```
switch(config)# int 1/1/1
switch(config-if)# apply policy RATE_LIMIT_HTTP in
switch(config-if)# exit
```

- Show the running configuration with RATE_LIMIT_HTTP applied:

```
switch# show running-config
Current configuration:
!
...
class ip HTTP_TRAFFIC
  10 match tcp any any eq 80
policy RATE_LIMIT_HTTP
  10 class ip HTTP_TRAFFIC action cir kbps 204800 exceed drop
interface 1/1/1
  apply policy RATE_LIMIT_HTTP in
```

```
switch# show running-config
Current configuration:
!
...
class ip HTTP_TRAFFIC
  10 match tcp any any eq 80
policy RATE_LIMIT_HTTP
  10 class ip HTTP_TRAFFIC action cir kbps 204800 exceed drop
interface 1/1/1
  apply policy RATE_LIMIT_HTTP in
```

Switches have finite (TCAM and other) hardware resources used in the application of ACLs and Classifier policies to packets being processed in switch hardware. ADC (analytics data collection) also consumes TCAM lookups. Take the considerations described in this chapter into account when deciding what ACL and classifier policy-related features to use at the same time.

TCAM lookups

TCAM lookups are a finite hardware resource used in the application of ACLs and policies to packets being processed in switch hardware. ADC (analytics data collection) also consumes TCAM lookups. There are a limited number of ACL and policy features that can be enabled at the same time. TCAM resources and lookups can be shown and monitored using command `show resources`.

There are two TCAM lookups available to use per line card. Each of these features uses one TCAM lookup when enabled. At most, two of these features can be enabled at the same time.

```
Ingress Port IPv6 ACL
Ingress Port MAC ACL
Ingress VLAN IPv4 and/or IPv6 ACL
Ingress VLAN MAC ACL
Ingress Routed VLAN IPv4 and/or IPv6 ACL
Ingress Port Policy with IPv4 and/or MAC classes
Ingress Port Policy with IPv6 classes
Ingress VLAN Policy with IPv4 and/or MAC classes
Ingress VLAN Policy with IPv6 classes
Ingress Routed Port Policy with IPv4 classes
Ingress Routed Port Policy with IPv6 classes
Ingress Routed VLAN Policy with IPv4 classes
Ingress Routed VLAN Policy with IPv6 classes
```

These features are not classifier-related but do use one TCAM lookup each:

```
IPv6 6in6 Tunnel Control Plane Policing
Ingress IPv4 VSX
Ingress IPv6 VSX
Ingress MAC VSX
Egress VSX
Ingress IPv4 and/or Analytics Data Collection (ADC)
Bidirectional Forwarding Detection (BFD)
Ingress Routed IPv4 and/or IPv6 Unicast Counters
Ingress Routed IPv4 and/or IPv6 Multicast Counters
Egress Routed IPv4 and/or IPv6 Unicast Counters
Egress Routed IPv4 and/or IPv6 Multicast Counters
Explicit Congestion Notification (ECN)
```

Port IPv4 ACLs use dedicated hardware and do not conflict with any of the features listed in this topic.

Matching precedence order

When a packet is matched by multiple TCAM Lookups with the same action, a precedence order is followed.

For example, if a packet matches an IPv6 ACL with a count action and a MAC ACL with a count action, the IPv6 count action takes precedence and the MAC ACL will not count the packet. However, if a packet matches both an ACL and a policy with count actions, both will be counted. Regardless of precedence, if a packet is to be dropped by a configured feature, it will be dropped. Ingress packets do not take precedence over egress packets nor do egress packets take precedence over ingress packets.

Applies only to deny, not permit: If a policy counts and mirrors a packet which also matches an ACL that counts the packet, the hitcounts for the ACL and policy will not increment. Mirrored packets should configure only one counter, in either the ACL or the policy, but not both.

The precedence order from highest to lowest is as follows:

```
Ingress Port IPv6 ACL
Ingress VLAN IPv6 ACL
Ingress Routed VLAN IPv6 ACL
Ingress Port IPv4 ACL
Ingress Routed VLAN IPv4 ACL
Ingress VLAN IPv4 ACL
Ingress Port MAC ACL
Ingress VLAN MAC ACL
Ingress Port Policy with IPv6 classes
Ingress Port Policy with IPv4 and/or MAC classes
Ingress VLAN Policy with IPv6 classes
Ingress VLAN Policy with IPv4 and/or MAC classes
Bidirectional Forwarding Detection (BFD)
Ingress IPv4 VSX
Ingress IPv6 VSX
Ingress MAC VSX
IPv6 Control Plane Policing
IPv4 Control Plane Policing
MAC Control Plane Policing
Ingress Routed IPv6 Port Policy
Ingress Routed IPv4 Port Policy
Ingress Routed IPv6 VLAN Policy
Ingress Routed IPv4 VLAN Policy
Ingress Routed IPv6 Unicast Counters
Ingress Routed IPv6 Multicast Counters
Ingress Routed IPv4 Unicast Counters
Ingress Routed IPv4 Multicast Counters
Ingress IPv6 Analytics Data Collection (ADC)
Ingress IPv4 Analytics Data Collection (ADC)
Egress VSX
Egress Routed IPv4 Port ACL
Egress Routed IPv6 Unicast Counters
Egress Routed IPv6 Multicast Counters
Egress Routed IPv4 Unicast Counters
Egress Routed IPv4 Multicast Counters
```

L4 port ranges

Any ACE or class entry that uses 'lt', 'gt', 'range', or port groups will attempt to use dedicated hardware resources called L4 port ranges. There are 24 available L4 port ranges.

L4 port ranges are only supported on these features:

```
Ingress IPv4 VLAN ACL
Routed-in IPv4 VLAN ACL
Ingress IPv4 Port MAC Policy
Ingress IPv4 VLAN MAC Policy
```



```
Ingress IPv4 Global MAC Policy
Routed-in IPv4 Port Policy
Routed-in IPv4 VLAN Policy
```

An L4 port range can be shared between any supported feature. Any unsupported feature may use more than one hardware entry to represent the range of L4 ports.

ACL and Policy hardware resource commands

show resources

```
show resources [<SLOT-ID>] [vsx-peer]
```

Description

Shows hardware resource consumption for the specified line modules or for all line modules. Resource data is updated every 10 seconds.

Hardware resource consumption information is shown for:

- TCAM entries
- Policers
- L4 Port Ranges

Parameter	Description
<SLOT-ID>	Specifies the VSF member on the 6300 switch and the member and slot of the line module on the 6400 switch. For example, on the 6400 switch, to specify the line module in member 1, slot 2, enter 1/2.
<SLOT-ID>	Specifies the member and slot of the line module. For example, to specify the line module in member 1, slot 2, enter 1/2.
vsx-peer	Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

Showing hardware resource consumption for line module 1/1:

```
switch# show resources 1/1

Resource Usage:

Mod  Description
     Resource                               Width  Used Reserved  Free
-----
1/1  Ingress IPv4 Port ACL
     High-Capacity TCAM/LPM Entries         2      0   262144
     MAC Control Plane Policing
     TCAM Entries                           2     16    256
     IPv4 Control Plane Policing
```

TCAM Entries	2	70	256
IPv6 Control Plane Policing			
TCAM Entries	2	72	*
IPv4 Unicast Route			
High-Capacity TCAM/LPM Entries	1	0	131072
IPv6 Unicast Route			
High-Capacity TCAM/LPM Entries	2	0	262144
IPv4 Multicast Route			
High-Capacity TCAM/LPM Entries	2	0	65536
IPv6 Multicast Route			
High-Capacity TCAM/LPM Entries	4	0	65536
Total			
TCAM Entries		158	512 49664
High-Capacity TCAM/LPM Entries		0	786432 258048
Policers		0	65536
Ingress L4 Port Ranges		0	24

* This feature shares reserved resources with the preceding feature.

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
8400	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

The high capacity TCAM supports up to 512k entries. The standard TCAM supports up to 12k entries. Previously, the high capacity TCAM was dedicated to **Ingress IPv4 Port ACLs**. It is now possible to reconfigure the high capacity TCAM to instead be used for one of two other IPv4 ingress ACL types.

- **Ingress IPv4 Port ACLs** (the default)
- **Ingress IPv4 VLAN ACLs**
- **Routed Ingress IPv4 VLAN ACLs**

This makes it possible for you to configure your TCAM capacity according to the type of IPv4 ACLs that you expect to use most heavily. It is expected that this type of configuration would be performed infrequently. Upon reboot, the high capacity TCAM is entirely reconfigured for the selected IPv4 ingress ACL type, causing the previous ACL type to now only use the standard TCAM.



If any of the ACLs of the type moved to the standard TCAM are too large to have their ACEs programmed in their entirety due to lack of space in the standard TCAM, such ACLs will be left unapplied.

Configuring the high capacity TCAM



Configuring the high capacity TCAM requires a reboot of either the switch chassis or all line modules so this configuration change should be scheduled for a maintenance window.

Performing the configuration

Issue the `system high-capacity-tcam access-list ip` command, selecting one of three IPv4 ingress ACL types:

- `port in` for Ingress IPv4 Port ACLs (the default)
- `vlan in` for Ingress IPv4 VLAN ACLs
- `vlan routed-in` for Routed Ingress IPv4 VLAN ACLs

For example, configuring Ingress IPv4 VLAN ACLs:

```
switch(config)# system high-capacity-tcam access-list ip vlan in  
This command changes how the high capacity TCAM is used and requires  
a reboot to complete. After reboot, any entries that were using the  
high capacity TCAM will be moved to lower capacity hardware and may  
no longer fit.  
  
Continue (y/n)? y  
switch(config)#
```

Showing the configuration before reboot. Notice the indicated Configured feature shows `Ingress IPv4 VLAN ACL` and the reminders to reboot.

```

switch# show system high-capacity-tcam
High-Capacity TCAM/LPM Usage:
Configured feature : Ingress IPv4 VLAN ACL
Default feature   : Ingress IPv4 Port ACL

Module      Current Feature
-----
1/1         Ingress IPv4 Port ACL (reboot required to apply new configuration)
1/4         Ingress IPv4 Port ACL (reboot required to apply new configuration)

```

Verifying the updated configuration

After either rebooting the entire switch or all of its line modules, verify that the updated configuration is now in effect. Notice how the selected ACL type `Ingress IPv4 VLAN ACLs` is now shown for all line modules and matches the indicated `Configured feature`:

```

switch# show system high-capacity-tcam
High-Capacity TCAM/LPM Usage:
Configured feature : Ingress IPv4 VLAN ACL
Default feature   : Ingress IPv4 Port ACL

Module      Current Feature
-----
1/1         Ingress IPv4 VLAN ACL
1/4         Ingress IPv4 VLAN ACL

```

The `show resources` command can be used to see that the `Ingress IPv4 VLAN ACL` is now assigned to the high capacity TCAM (bolded in this sample output):

```

switch# show resources 1/1

Resource Usage:

Mod  Description
    Resource                               Width  Used Reserved  Free
-----
1/1  Ingress IPv4 Port ACL
    TCAM Entries                           4      16      8192
Ingress IPv4 VLAN ACL
High-Capacity TCAM/LPM Entries         2      0 262144
MAC Control Plane Policing
    TCAM Entries                           2      18      256
IPv4 Control Plane Policing
    TCAM Entries                           2      80      256
IPv6 Control Plane Policing
    TCAM Entries                           2      78      *
Total
    TCAM Entries                           176     512  41472
    High-Capacity TCAM/LPM Entries         0      4096 1044480
    Policers                               0      65536
    Ingress L4 Port Ranges                  0      24
* This feature shares reserved resources with the preceding feature.

```

VSX sync requirements



In a VSX environment the high capacity TCAM configuration must be synchronized to the secondary switch and both switches (or all line cards in both switches) must be rebooted.

Running the `vsx-sync` command on the primary switch:

```
switch-primary(config)# vsx  
switch-primary(config-vsx)# vsx-sync hardware-high-capacity-tcam
```

Showing the high capacity TCAM configuration on the secondary switch. Notice the indicated Configured feature shows `Ingress IPv4 VLAN ACL` and the reminders to reboot.

```
switch-secondary(config)# show system high-capacity-tcam  
High-Capacity TCAM/LPM Usage:  
Configured feature : Ingress IPv4 VLAN ACL  
Default feature   : Ingress IPv4 Port ACL  
  
Module      Current Feature  
-----  
1/1         Ingress IPv4 Port ACL (reboot required to apply new configuration)  
1/4         Ingress IPv4 Port ACL (reboot required to apply new configuration)
```

After either rebooting the entire secondary switch or all of its line modules, verify that the updated configuration is now in effect. Notice how the selected ACL type `Ingress IPv4 VLAN ACLs` is now shown for all line modules and matches the indicated `Configured feature`:

```
switch-secondary# show system high-capacity-tcam  
High-Capacity TCAM/LPM Usage:  
Configured feature : Ingress IPv4 VLAN ACL  
Default feature   : Ingress IPv4 Port ACL  
  
Module      Current Feature  
-----  
1/1         Ingress IPv4 VLAN ACL  
1/4         Ingress IPv4 VLAN ACL
```

High capacity TCAM commands

system high-capacity-tcam access-list ip

```
system high-capacity-tcam access-list ip port in  
system high-capacity-tcam access-list ip vlan {in|routed-in}  
  
no system high-capacity-tcam access-list ip port in  
no system high-capacity-tcam access-list ip vlan {in|routed-in}
```

Description

Configures the high capacity TCAM for one of three types of IPv4 ingress ACLs. Full switch chassis reboot or reboot of every line module is required before the configuration change takes effect.

The `no` form of this command reverts the high capacity TCAM to its default of Ingress IPv4 Port ACLs.

Parameter	Description
port in	Selects Ingress IPv4 Port ACLs. This is the default.
vlan in	Selects Ingress IPv4 VLAN ACLs.
vlan routed-in	Selects Routed Ingress IPv4 VLAN ACLs.

Usage

- Upon reboot, the high capacity TCAM is entirely reconfigured for the selected IPv4 ingress ACL type, causing the previous ACL type to now only use the standard internal TCAM with its smaller capacity.



In a VSX environment the high capacity TCAM configuration must be synchronized to the secondary switch and both switches (or all line cards in both switches) must be rebooted before the new configuration takes effect on either switch.

Examples

Configuring the high capacity TCAM for Ingress IPv4 VLAN ACLs:

```
switch(config)# system high-capacity-tcam access-list ip vlan in
This command changes how the high capacity TCAM is used and requires
a reboot to complete. After reboot, any entries that were using the
high capacity TCAM will be moved to lower capacity hardware and may
no longer fit.

Continue (y/n)? y
switch(config)#
```

Reverting the high capacity TCAM to its default of Ingress IPv4 Port ACLs:

```
switch(config)# no system high-capacity-tcam access-list ip vlan in
This command changes how the high capacity TCAM is used and requires
a reboot to complete. After reboot, any entries that were using the
high capacity TCAM will be moved to lower capacity hardware and may
no longer fit.

Continue (y/n)? y
switch(config)#
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
8400	config	Administrators or local user group members with execution rights for this command.

show system high-capacity-tcam

show system high-capacity-tcam [commands]

Description

Shows the high capacity TCAM feature current configuration or the pending configuration change (if any) and the current high capacity TCAM feature configuration of each line module plus a reminder to reboot (when needed). The three possible high capacity TCAM feature names are displayed as follows (Ingress IPv4 Port ACL is the default):

```
Ingress IPv4 Port ACL
Ingress IPv4 VLAN ACL
Routed Ingress IPv4 VLAN ACL
```

Parameter	Description
commands	Changes the show command output to be in the form of CLI commands.

Examples

Configuring the high capacity TCAM for **Ingress IPv4 VLAN ACLs** and then showing the configuration before and after reboot. Notice the reminders for the required reboot.

```
switch(config)# system high-capacity-tcam access-list ip vlan in
This command changes how the high capacity TCAM is used and requires
a reboot to complete. After reboot, any entries that were using the
high capacity TCAM will be moved to lower capacity hardware and may
no longer fit.

Continue (y/n)? y
switch(config)#
switch(config)# show system high-capacity-tcam
High-Capacity TCAM/LPM Usage:
Configured feature : Ingress IPv4 VLAN ACL
Default feature    : Ingress IPv4 Port ACL

Module      Current Feature
-----
1/1         Ingress IPv4 Port ACL (reboot required to apply new configuration)
1/4         Ingress IPv4 Port ACL (reboot required to apply new configuration)

...
## after the reboot of the entire switch or all line modules ##

switch# show system high-capacity-tcam
High-Capacity TCAM/LPM Usage:
Configured feature : Ingress IPv4 VLAN ACL
Default feature    : Ingress IPv4 Port ACL

Module      Current Feature
-----
1/1         Ingress IPv4 VLAN ACL
1/4         Ingress IPv4 VLAN ACL
```

Showing the configuration change as commands:

```
switch# show system high-capacity-tcam commands
system high-capacity-tcam access-list ip vlan in
! not applied to module 1/1 - reboot required
! not applied to module 1/4 - reboot required
...
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
8400	Operator (>) or Manager (#)	Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Accessing Aruba Support

Aruba Support Services	https://www.arubanetworks.com/support-services/
Aruba Support Portal	https://asp.arubanetworks.com/
North America telephone	1-800-943-4526 (US & Canada Toll-Free Number) +1-408-754-1200 (Primary - Toll Number) +1-650-385-6582 (Backup - Toll Number - Use only when all other numbers are not working)
International telephone	https://www.arubanetworks.com/support-services/contact-support/

Be sure to collect the following information before contacting Support:

- 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

Other useful sites

Other websites that can be used to find information:

Airheads social forums and Knowledge Base	https://community.arubanetworks.com/
AOS-CX Switch Software Documentation Portal	https://www.arubanetworks.com/techdocs/AOS-CX/help_portal/Content/home.htm

Aruba Hardware Documentation and Translations Portal	https://www.arubanetworks.com/techdocs/hardware/DocumentationPortal/Content/home.htm
Aruba software	https://asp.arubanetworks.com/downloads
Software licensing	https://lms.arubanetworks.com/
End-of-Life information	https://www.arubanetworks.com/support-services/end-of-life/
Aruba Developer Hub	https://developer.arubanetworks.com/

Accessing Updates

You can access updates from the Aruba Support Portal or the HPE My Networking Website.

Aruba Support Portal

<https://asp.arubanetworks.com/downloads>

If you are unable to find your product in the Aruba Support Portal, you may need to search My Networking, where older networking products can be found:

My Networking

<https://www.hpe.com/networking/support>

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:

<https://support.hpe.com/portal/site/hpsc/aae/home/>

Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

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 subscribe to eNewsletters and alerts:

<https://asp.arubanetworks.com/notifications/subscriptions> (requires an active Aruba Support Portal (ASP) account to manage subscriptions). Security notices are viewable without an ASP account.

Warranty Information

To view warranty information for your product, go to <https://www.arubanetworks.com/support-services/product-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 <https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts>

Additional regulatory information

Aruba is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements, environmental data (company programs, product recycling, energy efficiency), and safety information and compliance data, (RoHS and WEEE). For more information, see <https://www.arubanetworks.com/company/about-us/environmental-citizenship/>.

Documentation Feedback

Aruba 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-switching@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.