AOS-CX 10.09 ACLs and Classifier Policies Guide

4100i, 6000, 6100 Switch Series

a Hewlett Packard Enterprise company

Published: February 2022
Edition: 2

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

Chapter 1
About this document

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 4100i Switch Series (JL817A, JL818A)
- Aruba 6000 Switch Series (R8N85A, R8N86A, R8N87A, R8N88A, R8N89A)
- Aruba 6100 Switch Series (JL675A, JL676A, JL677A, JL678A, JL679A)

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
`example-text`	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: ■ *<example-text>* ■ <example-text> ■ *example-text* ■ *example-text*	Identifies a placeholder—such as a parameter or a variable—that you must substitute with an actual value in a command or in code: ■ 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.

Convention	Usage
`[ ]`	Brackets. Indicates that the enclosed item or items are optional.
... or . . .	Ellipsis: ■ 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 4100i Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Always 1. This is not a modular switch, so there are no slots.
- *port*: Physical number of a port on the switch.

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

On the 6000 and 6100 Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Always 1. This is not a modular switch, so there are no slots.
- *port*: Physical number of a port on the switch.

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

Chapter 2
Access Control Lists

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 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 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. A given interface or LAG supports the application of a single ACL per type. ACLs can be applied to interfaces or LAGs as follows:

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

If you apply an ACL of a particular type that is already in use, the switch replaces the current ACL with the new ACL.

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 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
```

ACL and ACE-related tasks

Common ACL and ACE-related tasks are as follows:

Task	Command name	Example

Creating an IPv4 ACL	`access-list ip`	`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`

Creating an IPv6 ACL	`access-list ipv6`	`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`

Creating a MAC ACL	`access-list mac`	`access-list mac MY_MAC_ACL`
		`  10 permit any any appletalk vlan 40`
		`  20 deny any any any count`

Applying an IPv6 ACL to an interface	`apply access-list` (to interface or LAG)	`interface 1/1/1` `  apply access-list ipv6 MY_IPV6_ACL in`
Applying an IPv4 ACL to a LAG	`apply access-list` (to interface or LAG)	`interface lag 100` `  apply access-list ip MY_IP_ACL in`
Applying an IPv4 ACL to a VLAN	`apply access-list` (to VLAN)	`vlan 10` `  apply access-list ip MY_IP_ACL in`
Applying a MAC ACL to a VLAN	`apply access-list` (to VLAN)	`vlan 40` `  apply access-list mac MY_MAC_ACL in`
Removing application of an ACL from an interface	`apply access-list` (to interface or LAG)	`interface 1/1/1` `  no apply access-list ipv6 MY_IPV6_ACL in`
Removing application of an ACL from a VLAN	`apply access-list` (to interface VLAN)	`vlan 40` `  no apply access-list mac MY_MAC_ACL in`
Showing all ACLs	`show access-list`	`show access-list`
Showing all IPv6 ACLs	`show access-list`	`show access-list ipv6`
Showing all ACLs applied to interface `1/1/1`	`show access-list`	`show access-list interface 1/1/1`
Showing all ACLs applied to VLAN `10`	`show access-list`	`show access-list vlan 10`

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

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

```
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

On the 6000 and 6100 Switch Series, only the vrf named `default` is available. Replace any references to the `mgmt` or other VRFs with `default`.

On the 4100i Switch Series, only the vrf named `default` is available. Replace any references to the `mgmt` or other VRFs with `default`.

ACL application

ACLs can be applied as follows:

ACL type Direction	IPv4+6 In	MAC In
L2 interface (port)	Yes	Yes
L2 LAG	Yes	Yes
VLAN	Yes	Yes
Control plane (default VRF)	Yes	

The following match criteria is not supported. If this match criteria is attempted to be configured, an error message will be displayed and the action will not be completed.

```
TTL on IP ACLs
```

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

- 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:

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                          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
------------------------------------------------------------------------------
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                          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-1 tagged Appletalk traffic
           permit                          appletalk
           any
           any
           VLAN: 1
         4 deny                            any
           any
           any
           Hit-counts: enabled
------------------------------------------------------------------------------
MAC        MY_MAC_ACL2
         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-1 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 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>}]}
{any|<DST-IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[dscp <DSCP-SPECIFIER>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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>}]}
[{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>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>]  [fragment] [vlan <VLAN-ID>] [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>}]}
{any|<DST-IP-ADDRESS>[/{<PREFIX-LENGTH>|<SUBNET-MASK>}]}
[icmp-type {echo|echo-reply|<ICMP-TYPE-VALUE>}] [icmp-code <ICMP-CODE-VALUE>]
[dscp <DSCP-SPECIFIER>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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.

Parameter	Description
<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>*}]}	Specifies the source IPv4 address. ■ any - specifies any source IPv4 address. ■ *<SRC-IP-ADDRESS>* - specifies the source IPv4 host address. ◦ *<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. ■ any - specifies any destination IPv4 address. ■ *<DST-IP-ADDRESS>* - specifies the destination IPv4 host address. ◦ *<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. ■ 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. **NOTE:** Upon application of the ACL, ACEs with L4 port ranges may consume more than one hardware entry.
urg	Specifies matching on the TCP Flag: Urgent.
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.

Parameter	Description
`established`	Specifies matching on the TCP Flag: `Established connection`.
`[icmp-type {echo|echo-reply|<ICMP-TYPE-VALUE>}]`	Specifies the ICMP type. ■ `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: ■ `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)
`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.

Parameter	Description
`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.
`count`	Keeps the hit counts of the number of packets matching this ACE.
`log`	
`[<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                          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
```

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                          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 all TCP ephemeral ports
           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 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           >  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
```

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           >  1023
           any
        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
           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                          udp
           any
           172.16.1.0/255.255.255.0
        20 permit                          tcp
           172.16.2.0/255.255.0.0           >  1023
           any
        25 permit                          icmp
           172.17.1.0/255.255.0.0
        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 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                          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
```

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 `access-list ip` *<ACL-NAME>* 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>]}
{any|<DST-IP-ADDRESS>[/<PREFIX-LENGTH>]}
[dscp <DSCP-SPECIFIER>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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>}]}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>]
{any|<DST-IP-ADDRESS>[/<PREFIX-LENGTH>]}
[{eq|gt|lt} <PORT>|range <MIN-PORT> <MAX-PORT>]
[urg] [ack] [psh] [rst] [syn] [fin] [established]
[dscp <DSCP-SPECIFIER>] [ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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>]}
{any|<DST-IP-ADDRESS>[/<PREFIX-LENGTH>]}
[icmp-type {echo|echo-reply|<ICMP-TYPE-VALUE>}] [icmp-code <ICMP-CODE-VALUE>]
[dscp <DSCP-SPECIFIER>][ip-precedence <IP-PRECEDENCE-VALUE>]
[tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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.
<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>*]}	Specifies the source IPv6 address. ■ `any` - specifies any source IPv6 address. ■ *<SRC-IP-ADDRESS>* - specifies the source IPv6 host address. ◦ *<PREFIX-LENGTH>* - specifies the address bits to mask (CIDR subnet mask notation). Range: 1 to 128.
{any|*<DST-IP-ADDRESS>*[/*<PREFIX-LENGTH>*]}	Specifies the destination IPv6 address. ■ `any` - specifies any destination IPv6 address. ■ *<DST-IP-ADDRESS>* - specifies the destination IPv6 host address. ◦ *<PREFIX-LENGTH>* - specifies the address bits to mask (CIDR

Parameter	Description
	subnet mask notation). Range: 1 to 128.
`[{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. ■ `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. **NOTE:** Upon application of the ACL, ACEs with L4 port ranges may consume more than one hardware entry.
`[icmp-type {echo|echo-reply|<ICMP-TYPE-VALUE>}]`	Specifies the ICMP type. ■ `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: ■ `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)

Parameter	Description
	■ `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)
`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.
`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.
`count`	Keeps the hit counts of the number of packets matching this ACE.
`log`	
`[`*`<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 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                          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
```

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 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 all TCP ephemeral ports
           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 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 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
```

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 `access-list ipv6 <ACL-NAME>` 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|aarp|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|aarp|appletalk| ... |wake-on-lan|*<NUMERIC-ETHERTYPE>*	Specifics 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: ■ `any` - any EtherType. ■ *<NUMERIC-ETHERTYPE>* - the numerical EtherType protocol number. Range: 0x600 to 0xffff. ■ One of these EtherType protocol name keywords: ◦ `aarp` ◦ `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`

Parameter	Description
`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. **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`	

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                          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: 40
         40 deny                            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                          ipv6
           1122.3344.5566/ffff.ffff.0000
           any
        20 permit                          any
           aaaa.bbbb.cccc
           1111.2222.3333
           QoS Priority Code Point: 4
        30 Permit all vlan-40 tagged Appletalk traffic
           permit                          appletalk
           any
           any
           VLAN: 40
        40 deny                            any
           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                          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
        40 deny                            any
           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
           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
           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                          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
        40 deny                            any
           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                          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` The `access-list mac <ACL-NAME>` 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           >  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
```

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: ■ 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
`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.

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

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
`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 interface or LAG once. Therefore, using the `apply access-list` command on an interface or LAG with an already-applied ACL of the same type 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.

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_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)#
```

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 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 stack members regardless of whether a VLAN member exists on any specific stack member.

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] }
```

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.

Examples

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.

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] [commands] [configuration]
```

Syntax that filters by the named ACL:

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

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 IPv4 or IPv6 ACLs applied to interface VLANs.

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

Examples

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                          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
           syn
           ack
           dscp 10
        40 deny                            any
           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                          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
           syn
           ack
           dscp 10
        40 deny                            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                          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::2:1/128               >  1023
           any
        30 permit                          tcp
           2001:2011::1/64
        40 deny                            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                          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 deny                            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]
```

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.

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 Comment
           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
```

```
           172.16.2.0/16                    >  1023
           any
        30 permit                           tcp
           172.26.1.0/24
           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 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::2:1/128               >  1023
           any
        30 permit                           tcp
           2001:2011::1/64
        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]}
```

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.

Usage

- ACL hit counts are aggregated across all:
  - physical interfaces to which the ACL is applied to on ingress,
  - VLANs to which the ACL is applied to on 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>`

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
`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.

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>`

Description

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

Parameter	Description
<FEATURE>	Specifies a feature. For example, `aaa`.

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 classifier-related capacities on the 4100i, 6000, or 6100:

```
switch# show capacities classifier

System Capacities: Filter Classifier
Capacities Name                                                                    Value
-------------------------------------------------------------------------------------
Maximum number of Access Control Entries configurable in a system                   4096
Maximum number of Access Control Lists configurable in a system                      512
Maximum number of class entries configurable in a system                            4096
Maximum number of classes configurable in a system                                   512
Maximum number of entries in an Access Control List                                 1024
Maximum number of entries in a class                                                1024
```

```
Maximum number of entries in a policy                                        64
Maximum number of classifier policies configurable in a system              512
Maximum number of policy entries configurable in a system                   4096
```

Showing all available capacities on the 4100i, 6000, or 6100:

```
switch# show capacities

System Capacities:
Capacities Name                                                            Value
-------------------------------------------------------------------------------
Maximum number of Access Control Entries configurable in a system          4096
Maximum number of Access Control Lists configurable in a system             512
Maximum number of class entries configurable in a system                   4096
Maximum number of classes configurable in a system                          512
Maximum number of entries in an Access Control List                        1024
Maximum number of entries in a class                                       1024
Maximum number of entries in a policy                                        64
Maximum number of classifier policies configurable in a system              512
Maximum number of policy entries configurable in a system                  4096
Maximum number of clients supported for tracking the IP address in the system  128
Maximum number of dynamic VLANs that can be allowed using MVRP              256
Maximum number of nexthops per IP ECMP group                                  1
Maximum number of IP neighbors (IPv4+IPv6) supported in the system          1024
Maximum number of IPv4 neighbors(# of ARP entries) supported in the system  1024
Maximum number of IPv6 neighbors(# of ND entries) supported in the system    512
Maximum number of L2 MAC addresses supported in the system                  8192
Maximum number of L3 Groups for IP Tunnels and ECMP Groups                    1
Maximum number of L3 Destinations for Routes, Nexthops in Tunnels and ECMP groups
1024
Maximum number of configurable LAG ports                                      8
Maximum number of members supported by a LAG port                             8
Maximum number of VLANs across ports allowed in loop-protect                3328
Maximum number of IGMP/MLD groups supported                                  512
Maximum number of IGMP/MLD snooping groups supported                         512
Maximum number of Mirror Sessions configurable in a system                     4
Maximum number of enabled Mirror Sessions in a system                          4
Maximum number of mstp instances configurable in a system                     16
Maximum number of Clients that can be authenticated on a port                 32
Maximum number of Device Profiles allowed to be created on the system          8
Maximum number of Port Access Roles allowed to be created on the system       32
Maximum number of MAC Address that can be authorized on a port                32
Maximum number of Port Access Role VLAN IDs allowed to be created on the system 50
Maximum number of Port Access Role VLAN names allowed to be created on the system
 50
Maximum number of RBAC rules per user group                                 1024
Maximum number of RPVST VLANs configurable on the system                      16
Maximum number of RPVST VPORTs supported in a system                         512
Maximum number of SVIs supported in the system                                16
Maximum number of routes (IPv4+IPv6) on the system                           512
Maximum number of IPv4 routes on the system                                  512
Maximum number of IPv6 routes on the system                                  512
Maximum number of VLANs supported in the system                              512
```

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>`

Description

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

Parameter	Description
<FEATURE>	Specifies the feature, for example `aaa` for which to display capacities, values, and status. Required.

Examples

Showing the system capacities status for all features:

```
switch# show capacities-status
System Capacities Status
Capacities Status Name                                                    Value
Maximum
-------------------------------------------------------------------------------
--
Number of Access Control Entries currently configured                       0
4096
Number of Access Control Lists currently configured                         0
512
Number of class entries currently configured                                0
4096
Number of classes currently configured                                      0
512
Number of policies currently configured                                     0
512
Number of policy entries currently configured                               0
4096
Number of dynamic VLANs currently learnt using MVRP                         0
256
Number of IP neighbor (IPv4+IPv6) entries                                   1
1024
Number of IPv4 neighbor(ARP) entries                                        1
1024
Number of IPv6 neighbor(ND) entries                                         0
512
Number of L3 Groups for IP Tunnels and ECMP Groups currently configured     0
1
Number of L3 Destinations for Routes, Nexthops in ECMP groups and Tunnels
  currently configured                                                      0
1024
Number of Mirror Sessions currently configured                              0
4
Number of Mirror Sessions currently enabled                                 0
4
Number of mstp instances currently configured                               0
16
Number of RPVST VLANs currently configured                                  0
16
Number of routes (IPv4+IPv6) currently configured                           1
512
Number of IPv4 routes currently configured                                  1
512
Number of IPv6 routes currently configured                                  0
512
Number of VLANs currently configured                                        1
512
```

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.

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 (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:

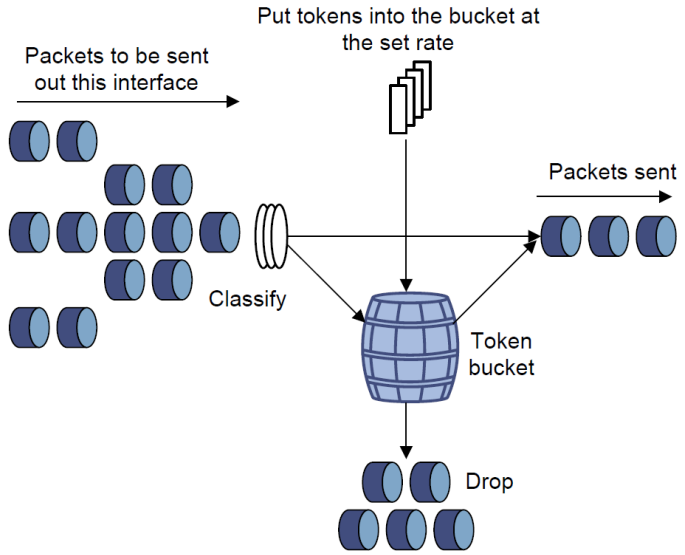
- Frame ingress VLAN ID
- Source and/or destination IPv4, or IPv6 address
- Layer 2 (EtherType) and Layer 3 (IP) protocol
- Layer 4 application ports

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:

- **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.
- **Class of Service (CoS):** Queuing can be achieved through the cos action field which will remark the cos value in case of tagged packets and queuing of both tagged and untagged packets.

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.

Based on these parameters, packets are dropped when traffic exceeds the bandwidth limit (CIR).

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>} 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:

Policy type Direction	IPv4 In	IPv6 In
L2 interface (port)	Yes	Yes
L2 LAG	Yes	Yes
VLAN	Yes	Yes

Port policies and port-access client policies cannot be configured at the same time.

apply policy (config)

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

Description

Applies a policy to the global config context.

Only one policy can be globally applied at a time. Applying a policy globally again, replaces the previous globally applied policy.

The `no` form of this command removes application of the global policy.

Parameter	Description
<POLICY-NAME>	Specifies the policy to apply.
`in`	Selects the inbound (ingress) traffic direction.

Examples

Applying policy global1 to the global config context:

```
switch(config)# apply policy global1 in
```

Removing application of policy global1 from the global config context:

```
switch(config)# no apply policy global1 in
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
4100i 6000 6100	config	Administrators or local user group members with execution rights for this command.

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

Context config-if, config-lag-if:

```
apply policy <POLICY-NAME> {in} [per-interface]
no apply policy <POLICY-NAME> {in} [per-interface]
```

Context config-vlan:

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

Description

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

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

Parameter	Description
<POLICY-NAME>	Specifies the policy to apply.
`in`	Selects the inbound (ingress) traffic direction.
`per-interface`	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.

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 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
```

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
```

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`
		Administrators or local user group members with execution rights for this command.

class copy

`class {ip|ipv6} <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} *<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                          L3 Protocol
           Source IP Address               Source L4 Port(s)
           Destination IP Address          Destination L4 Port(s)
           Additional Parameters
------------------------------------------------------------------------------
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
```

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>] [ip-precedence <IP-PRECEDENCE-VALUE>]
  [tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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>] [ip-precedence <IP-PRECEDENCE-VALUE>]
  [tos <TOS-VALUE>]  [fragment] [vlan <VLAN-ID>] [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>] [ip-precedence <IP-PRECEDENCE-VALUE>]
  [tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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.

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.
<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. ■ any - specifies any source IPv4 address. ■ *<SRC-IP-ADDRESS>* - specifies the source IPv4 host address. ○ *<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. ■ any - specifies any destination IPv4 address. ■ *<DST-IP-ADDRESS>* - specifies the destination IPv4 host address. ○ *<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. ■ 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.

Parameter	Description
	■ `range` *<MIN-PORT> <MAX-PORT>* - specifies the Layer 4 port range.
`urg`	Specifies matching on the TCP Flag: Urgent.
`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>*	Specifies the Differentiated Services Code Point (DSCP), either a numeric *<DSCP-VALUE>* (0 to 63) or one of these keywords: ■ `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

Parameter	Description
	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: 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)
`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.
`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

Parameter	Description
	`no` 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                           icmp
           any
           any
        20 ignore                          udp
           any
           any
        30 match                           tcp
           192.168.0.1
           192.168.0.2
```

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>] [ip-precedence <IP-PRECEDENCE-VALUE>]
  [tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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>] [ip-precedence <IP-PRECEDENCE-VALUE>]
  [tos <TOS-VALUE>]  [fragment] [vlan <VLAN-ID>] [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>] [ip-precedence <IP-PRECEDENCE-VALUE>]
  [tos <TOS-VALUE>] [fragment] [vlan <VLAN-ID>] [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
<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.
<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 IPv6 address. ■ any - specifies any source IPv6 address. ■ *<SRC-IP-ADDRESS>* - specifies the source IPv4 host address. ◦ *<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. ■ any - specifies any destination IPv6 address. ■ *<DST-IP-ADDRESS>* - specifies the destination IPv6 host address. ◦ *<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. ■ 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>* -

Parameter	Description
	specifies the Layer 4 port range.
`urg, ack, psh, rst, syn, fin, established`	These TCP flag matching parameters are not supported.
`dscp` *<DSCP-SPECIFIER>*	Specifies the Differentiated Services Code Point (DSCP), either a numeric *<DSCP-VALUE>* (0 to 63) or one of these keywords: ■ `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)

Parameter	Description
	■ `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)
`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.
`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 `no` 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                          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
```

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                         L3 Protocol
           Source IP Address              Source L4 Port(s)
           Destination IP Address         Destination L4 Port(s)
           Additional Parameters
-------------------------------------------------------------------------------
IPv6       MY_IPV6_CLASS
        10 match                          any
           any
           1020::
        20 ignore                         udp
           any
           any
```

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                         L3 Protocol
           Source IP Address              Source L4 Port(s)
           Destination IP Address         Destination L4 Port(s)
           Additional Parameters
-------------------------------------------------------------------------------
IPv6       MY_IPV6_CLASS
        20 ignore                         udp
           any
           any
```

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	`config` The `class ipv6 <CLASS-NAME>` command takes you into the `config-class-ipv6` command context where you enter the class entries.	Administrators or local user group members with execution rights for this command.

class resequence

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

Description

Resequence numbering in an IPv4, or IPv6 class.

Parameter	Description
{ip|ipv6} *<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
```

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>} 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>}`	Specifies either `all` classes be reset or specifies the type (`ip` for IPv4, or `ipv6` for IPv6) 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

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.
`global`	Selects the globally applied policy.

Examples

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} <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 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}` *<CLASS-NAME>*	Specifies a type of class, `ip` for IPv4, `ipv6` for IPv6.
<REMARK-ACTIONS>	Remark actions can be any of the following options: `{cos` *<COS-VALUE>* `| ip-precedence` *<IP-PRECEDENCE_VALUE>* `| dscp` *<DSCP-VALUE>*} where:
`cos` *<COS-VALUE>*	Specifies the Class of Service (CoS) value.

Parameter	Description
`ip-precedence <IP-PRECEDENCE-VALUE>`	Specifies the numeric IP precedence value. Range: 0 to 7.
`dscp <DSCP-VALUE>`	Specifies a Differentiated Services Code Point (DSCP) value. Enter either a numeric value (0 to 63) or a keyword as follows: ■ `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)
<POLICE-ACTIONS>	Police actions can be the following `{cir <RATE-BPS> exceed}` where:
`cir kbps <RATE-KBPS>`	Specifies a Committed Information Rate value in Kilobits per second. Range: 1 to 4294967295.
`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.

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 `policy` command takes you into the `config-policy` 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.
<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] [<CLASS-NAME>] [commands] [configuration]
```

Description

Shows class configuration information.

All parameters are optional.

Parameter	Description
`[ip | ipv6]`	Selects the class type for the display: `ip` for IPv4, `ipv6` for IPv6.
<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.

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]
```

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

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

Syntax that filters by the named policy:

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

Syntax that filters by the globally applied policy:

```
show policy global [commands] [configuration]
```

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

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

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

```
show policy hitcounts global
```

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, the statistics will be aggregated. 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
`interface <IF-NAME>`	Specifies the interface name.
`vlan <VLAN-ID>`	Specifies the VLAN.
`in`	Selects the inbound (ingress) traffic direction.
`<POLICY-NAME>`	Specifies the policy name.
`commands`	Causes the policy definition to be shown as the commands and parameters used to create it rather than in tabular form.
`configuration`	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.
`global`	Selects the globally applied policy.
`hitcounts`	Selects the policy hit counts (statistics).

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
kbps conform ]
                  -  10 match tcp any any ack
                  0  20 match icmpv6 1000::10 any count
```

Showing policy hit counts (statistics) for a policy applied everywhere (with 1/1/4 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  [ 0
kbps conform ]
                  -  10 match tcp any any ack
                  0  20 match icmpv6 1000::10 any count

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
kbps conform ]
                  -  10 match tcp any any ack
                  0  20 match icmpv6 1000::10 any count

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
kbps conform ]
                  -  10 match tcp any any ack
                  0  20 match icmpv6 1000::10 any count
```

```
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  [ 0
kbps conform ]
                   -  10 match tcp any any ack
                   0  20 match icmpv6 1000::10 any count
...
```

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
kbps conform ]
                   -  10 match tcp any any ack
                   0  20 match icmpv6 1000::10 any count
```

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

```
switch# show policy hitcounts My_Policy vlan 10
Statistics for Policy My_Policy:

vlan 10,20-30 (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
kbps conform ]
                   -  10 match tcp any any ack
                   0  20 match icmpv6 1000::10 any count
```

Command History

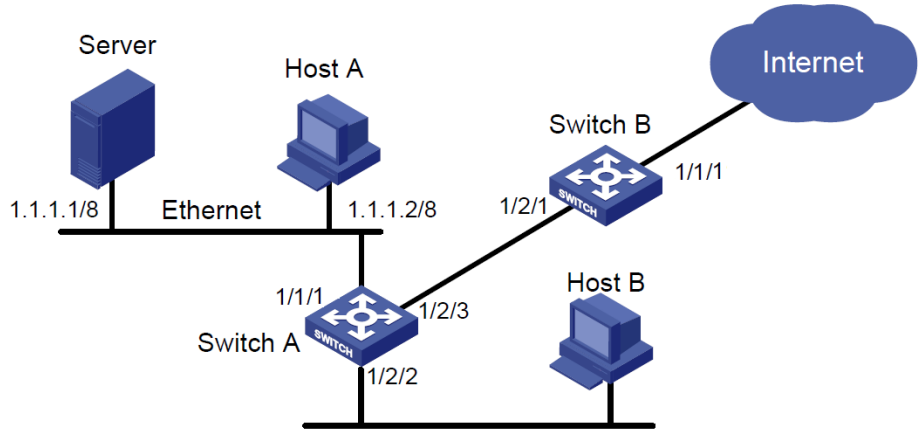
Release	Modification
10.08	Added `[per-interface]` information. Updated examples.
10.07 or earlier	--

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
```

9. 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
```

10. 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
```

11. 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
```

TCAM lookups

TCAM lookups are a finite hardware resource used in the application of ACLs and policies (including port access 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`.

In the following TCAM lookup lists, "IP" means both IPv4 and IPv6.

There are 32 TCAM lookups available to use for these features. Each of these features uses one TCAM lookup when enabled.

```
Ingress Port IP ACL
Ingress Port MAC ACL
Ingress Port Policy
Ingress Port Access Client Policy
Ingress VLAN IP ACL
Ingress VLAN MAC ACL
Ingress VLAN Policy
Ingress Global Policy
Port Access Client Policy
```

This features is not classifier related but uses one lookup:

```
Ingress CPURX
```

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 Policy with an action to change DSCP to AF11 and a MAC policy with an action to change DSCP to AF12, the MAC DSCP action takes precedence and the DSCP of the packet will change to AF12, given that the precedence of a IPv6 Policy is higher than the precedence of a MAC Policy. Count is an exception in that if a packet matches an IPv4 ACL, MAC ACL, and a policy with count actions, all the counters will increment.

The precedence order from highest to lowest is as follows:

```
Meter Actions:
Port Access Client Policy
Port Policy
VLAN Policy
Ingress Global Policy

QoS Actions:
```

```
Port Access Client Policy Remark
Ingress Port Policy Remark
Ingress VLAN Policy Remark
Ingress Global Policy Remark
QoS DSCP Map Entry
QoS COS Map Entry
QoS Port Config
MAC Port ACL Logging
IP Port ACL Logging
MAC VLAN ACL Logging
IP VLAN ACL Logging
```

L4 port ranges

Any ACE that uses 'lt', 'gt', 'range', or port groups may attempt to use dedicated hardware resources called L4 port ranges. There are 60 available L4 port ranges. An L4 port range can be shared between any supported feature.

Context group selectors

Context group selectors are a limited hardware resource that are required for applying ACLs and classifier policies. The selectors enable the application of an ACL or classifier policy to multiple instances of the same context (for example, ports on a line card or VLANs) without consuming additional resources.

There are a limited number of available context group selectors for each context group (Ingress Ports, Ingress VLANs, Egress Ports, Egress VLANs).

IP ACLs require two selectors that are allocated together; one selector for each address family (IPv4 and IPv6).

Context group selectors work on a first-come-first-served basis. IP ACLs and Classes require two selectors that are allocated together; one selector for each address family (IPv4 and IPv6). Once all the group selectors for a context group have been used, no new application type of ACL or classifier policy for the context group can be applied. For example, if an existing configuration has a MAC ACL, IP ACL, and classifier policy applied on ingress to ports, a policy cannot be applied to a port in the routed-in direction .

Context group selector consumption and availability is as follows:

Type	Selectors
Ingress Port MAC ACL	1
Ingress Port IP ACL	2
Ingress Port Policy	1
Ingress Routed Port Policy	1
Available Ingress Port Selectors	4
Ingress VLAN MAC ACL	1
Ingress VLAN IP ACL	2

Type	Selectors
Routed-Ingress VLAN IP ACL	2
Ingress VLAN Policy	1
Ingress Routed VLAN Policy	1
Available Ingress VLAN Selectors	4
Egress Port MAC ACL	1
Egress Port IP ACL	2
Egress Port Policy	1
Available Egress Port Selectors	5
Egress VLAN MAC ACL	1
Egress VLAN IP ACL	2
Routed-Egress VLAN IP ACL	2
Egress VLAN Policy	1
Available Egress VLAN Selectors	5

ACL and Policy hardware resource commands

show resources

```
show resources
```

Description

Shows hardware resource consumption. Resource data is updated every 10 seconds.

Hardware resource consumption information is shown for:

- TCAM entries
- TCAM lookups
- Policers

Usage

The widths for show resources can have features combined (IPv4 + IPv6) into one TCAM lookup. Therefore, the table widths for each ACL/classifier policy type are variable depending on what is applied. For example:

```
"Ingress IP Port ACL" = Ingress v4 Port ACLs + Ingress v6 Port ACLs
                      = 1 TCAM entry + 4 TCAM entries
                      = 5 TCAM entries
```

Widths per feature are as follows:

```
IPv4 ACL   2
MAC ACL    2
IPv6 ACL   4
IPv4 Class 2
IPv6 Class 4
```

Examples

Showing hardware resource consumption on a 6100 switch:

```
switch# show resources

Resource Usage:

Mod  Description
        Resource                                    Used Reserved    Free
-------------------------------------------------------------------------
1/1  Ingress IPv4 VLAN ACL Lookup
        Ingress TCAM Entries                          4      128
     Ingress IPv6 VLAN ACL Lookup
        Ingress TCAM Entries                          8      128
     Ingress IP CPURX Lookup
        Ingress TCAM Entries                        126      128
        Ingress Policers                             19
     Ingress IP Port Policy Lookup
        Ingress TCAM Entries                          2      128
     Ingress IP VLAN Policy Lookup
        Ingress TCAM Entries                         12      128
     Total
        Ingress TCAM Entries                        152      640    3448
        Ingress Lookups                               5               27
        Ingress Policers                             19             2029
```

Command History

Release	Modification
10.07 or earlier	--

Command Information

Platforms	Command context	Authority
4100i 6000 6100	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	https://www.arubanetworks.com/techdocs/hardware/DocumentationPortal/Content/home.htm

Portal	
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.