show j - show o

show jumbo-frame reservation

To view whether jumbo frames are enabled for all interfaces, use the **show jumbo-frame reservation** command.

show jumbo-frame reservation

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Jumbo frame reservation is enabled whenever you increase the MTU for any interface over 1500. It is automatically disabled when you return all MTUs to 1500 or lower.

Examples

The following is sample output from the **show jumbo-frame reservation** command when jumbo frame support is enabled:

```
> show jumbo-frame-reservation
Jumbo Frame Support is currently enabled
```

show kernel

To display information that the Linux brctl utility provides that you can use for debugging, use the **show kernel** command.

show kernel {**process** | **bridge** [**mac-address** *bridge_name*] | **cgroup-controller** [**cpu** | **cpuset** | **memory**] [**detail**] | **ifconfig** | **module**}

Syntax Description

bridge [**mac-address** *bridge_name*]	Displays the Linux tap bridges, their member ports, and the MAC addresses that have been learned at each port (including remote MAC addresses) that you can use for debugging. You can use the **mac-address** keyword to view MAC address details about a specific bridge. Use the command without the keyword to see the available bridge names, such as br0.
cgroup-controller [**cpu** | **cpuset** | **memory**] [**detail**]	Displays the cgroup-controller statistics. The **cpu**, **cpuset** and **memory** keywords allow you to filter the cgroup-controller statistics as per your requirements. Use the detail keyword to see extra information.
ifconfig	Displays the tap and bridge interface statistics.
module	Displays the modules that are installed and running.
process	Displays the current status of the active kernel processes running on the device.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

This command displays statistics for the various processes running on the kernel.

Examples

The following example displays output from the **show kernel process** command:

```
> show kernel process
PID PPID PRI NI      VSIZE      RSS      WCHAN STAT  RUNTIME COMMAND
  1    0  16  0     991232      268 3725684979    S       78 init
  2    1  34 19          0        0 3725694381    S        0 ksoftirqd/0
  3    1  10 -5          0        0 3725736671    S        0 events/0
  4    1  20 -5          0        0 3725736671    S        0 khelper
  5    1  20 -5          0        0 3725736671    S        0 kthread
  7    5  10 -5          0        0 3725736671    S        0 kblockd/0
  8    5  20 -5          0        0 3726794334    S        0 kseriod
 66    5  20  0          0        0 3725811768    S        0 pdflush
 67    5  15  0          0        0 3725811768    S        0 pdflush
 68    1  15  0          0        0 3725824451    S        2 kswapd0
 69    5  20 -5          0        0 3725736671    S        0 aio/0
171    1  16  0     991232       80 3725684979    S        0 init
172  171  19  0     983040      268 3725684979    S        0 rcS
201  172  21  0    1351680      344 3725712932    S        0 lina_monitor
202  201  16  0 1017602048   899932 3725716348    S      212 lina
203  202  16  0 1017602048   899932          0    S        0 lina
```

```
204  203  15  0 1017602048   899932          0    S        0 lina
205  203  15  0 1017602048   899932 3725712932    S        6 lina
206  203  25  0 1017602048   899932          0    R 13069390 lina
>
```

The following table explains each field.

Table 1: show kernel process Fields

Field	Description
PID	The process ID.
PPID	The parent process ID.
PRI	The priority of the process.
NI	The nice value, which is used in priority computation. The values range from 19 (nicest) to -19 (not nice to others),
VSIZE	The virtual memory size in bytes.
RSS	The resident set size of the process, in kilobytes.
WCHAN	The channel in which the process is waiting.
STAT	The state of the process: • R—Running • S—Sleeping in an interruptible wait • D—Waiting in an uninterruptible disk sleep • Z—zombie • T—Traced or stopped (on a signal) • P—Paging
RUNTIME	The number of jiffies that the process has been scheduled in user mode and kernel mode. The runtime is the sum of utime and stime.
COMMAND	The process name.

The following example displays output from the **show kernel module** command:

```
> show kernel module

Module                  Size  Used by    Tainted: P
cpp_base              861808  2
kvm_intel              44104  8
kvm                   174304  1 kvm_intel
msrif                   4180  0
tscsync                 3852  0
```

The following example displays output for the **show kernel ifconfig** command:

```
> show kernel ifconfig

br0       Link encap:Ethernet  HWaddr 42:9E:B8:6C:1F:23
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1708 (1.6 KiB)  TX bytes:0 (0.0 B)

br1       Link encap:Ethernet  HWaddr 6A:03:EC:BA:89:26
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.255.255.255
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

tap0      Link encap:Ethernet  HWaddr 6A:0C:48:32:FE:F4
          inet addr:127.0.2.2  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:148 errors:0 dropped:0 overruns:0 frame:0
          TX packets:186 errors:0 dropped:13 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:10320 (10.0 KiB)  TX bytes:12452 (12.1 KiB)

tap1      Link encap:Ethernet  HWaddr 8E:E7:61:CF:E9:BD
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:259 errors:0 dropped:0 overruns:0 frame:0
          TX packets:187 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:19368 (18.9 KiB)  TX bytes:14638 (14.2 KiB)

tap2      Link encap:Ethernet  HWaddr 6A:03:EC:BA:89:26
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

tap3      Link encap:Ethernet  HWaddr 42:9E:B8:6C:1F:23
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:187 errors:0 dropped:0 overruns:0 frame:0
          TX packets:256 errors:0 dropped:3 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:14638 (14.2 KiB)  TX bytes:19202 (18.7 KiB)

tap4      Link encap:Ethernet  HWaddr 6A:5C:60:BC:9C:ED
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

The following example displays output from the **show kernel bridge** command:

```
> show kernel bridge

bridge name     bridge id               STP enabled     interfaces
br0             8000.000000040001       no              tap1
                                                        tap3
br1             8000.84b261b192bd       no              tap2
                                                        tap4
                                                        tap5
```

The following example displays output from the **show kernel bridge mac-address** command:

```
> show kernel bridge mac-address br1

port no    mac addr       is local?   ageing timer
 1    00:21:d8:cb:dc:f7    no          12.93
 3    00:22:bd:d8:7d:da    no          12.93
 2    26:d2:9f:51:a4:90    yes          0.00
 1    4e:a4:e0:73:1f:ab    yes          0.00
 3    52:04:38:3d:79:c0    yes          0.00
```

Related Commands

Command	Description
show module	Shows information about the installed modules in the device.

show lacp

To display EtherChannel LACP information such as traffic statistics, system identifier, and neighbor details, enter this command.

show lacp {*channel_group_number* {**counters** | **internal** [**detail**] | **neighbor** [**detail**]} | **neighbor** [**detail**] | **sys-id**}

Syntax Description

channel_group_number	Specifies the EtherChannel channel group number, between 1 and 48, and only shows information about this channel group.
counters	Shows counters for the number of LACPDUs and markers sent and received.
detail	Shows additional detail for the item.
internal	Shows internal information.
neighbor	Shows neighbor information.
sys-id	Shows the LACP system ID.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show lacp sys-id** command:

```
> show lacp sys-id
32768,001c.c4e5.cfee
```

The following is sample output from the **show lacp counters** command:

```
> show lacp counters

              LACPDUs         Marker      Marker Response    LACPDUs
Port       Sent   Recv     Sent   Recv     Sent    Recv      Pkts Err
---------------------------------------------------------------------
Channel group: 1
Gi3/1      736    728      0      0        0       0         0
Gi3/2      739    730      0      0        0       0         0
Gi3/3      739    732      0      0        0       0         0
```

The following is sample output from the **show lacp internal** command:

```
> show lacp internal

Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
```

```
Channel group 1
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
--------------------------------------------------------------------------------
Gi3/1     SA      bndl      32768         0x1       0x1     0x302       0x3d
Gi3/2     SA      bndl      32768         0x1       0x1     0x303       0x3d
Gi3/3     SA      bndl      32768         0x1       0x1     0x304       0x3d
```

The following is sample output from the **show lacp neighbor** command:

```
> show lacp neighbor

Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1 neighbors

Partner's information:
          Partner Partner    LACP Partner  Partner   Partner  Partner     Partner
Port      Flags   State      Port Priority Admin Key Oper Key Port Number Port State
--------------------------------------------------------------------------------
Gi3/1     SA      bndl       32768         0x0       0x1      0x306       0x3d
Gi3/2     SA      bndl       32768         0x0       0x1      0x303       0x3d
Gi3/3     SA      bndl       32768         0x0       0x1      0x302       0x3d
```

Related Commands

Command	Description
show port-channel	Displays EtherChannel information in a detailed and one-line summary form. This command also displays the port and port-channel information.
show port-channel load-balance	Displays port-channel load-balance information along with the hash result and member interface selected for a given set of parameters.

show lacp cluster

To show the cLACP system MAC and ID, use the **show lacp cluster** command

show lacp cluster {**system-mac** | **system-id**}

Syntax Description

system-mac	Shows the system ID and whether it was auto-generated or entered manually.
system-id	Shows the system ID and priority.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show lacp cluster system-mac** command:

```
> show lacp cluster system-mac
lacp cluster system MAC is automatically generated: a300.010a.010a.
```

The following is sample output from the **show lacp cluster system-id** command:

```
> show lacp cluster system-id
5    ,a300.010a.010a
```

show last-upgrade status

To show information about the status of the last system software upgrade, use the **show last-upgrade status** command.

show last-upgrade status

Command History

Release	Modification
6.7	This command was introduced.

Example

The following example shows that the last upgrade was successful. In actual output, x.y.0 would be replaced by a real version number.

```
> show last-upgrade status
Upgrade from 6.7.0 to x.y.0 was successful.
Time started: Tue Dec  3 23:50:31 UTC 2020
```

The following example shows that the last upgrade was canceled. In actual output, x.y.0 would be replaced by a real version number.

```
> show last-upgrade status
Upgrade from 6.7.0 to x.y.0 failed.
Time started: Tue Dec  3 23:50:31 UTC 2020
Cancel Upgrade was successful.
```

Related Commands

Command	Description
show upgrade	Shows information on the current system software upgrade.
upgrade	Cancel, revert, or retry a system software upgrade.

show lisp eid

To view the EID table, use the **show lisp eid** command.

show lisp eid [**site-id** *id*]

Syntax Description

site-id *id*	View only EIDs for a particular site.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The device maintains an EID table that correlates the EID and the site ID.

Examples

The following is sample output from the **show lisp eid** command:

```
> show lisp eid
LISP EID       Site ID
10.44.33.105        2
10.44.33.201        2
192.168.11.1        4
192.168.11.2        4
```

Related Commands

Command	Description
clear cluster info flow-mobility counters	Clears the flow mobility counters.
clear lisp eid	Removes EIDs from the ASA EID table.
show cluster info flow-mobility counters	Shows flow mobility counters.
show conn	Shows traffic subject to LISP flow-mobility.
show service-policy	Shows the service policy.

show lldp

To display Link Layer Discovery Protocol (LLDP) status for an interface, use the **show lldp** command.

show lldp { **neighbors** | **statistics** | **status** } *interface_id*

Syntax Description

interface_id	Specifies the interface ID.
neighbors	Shows if LLDP neighborship is established.
statistics	Shows the LLDP statistics.
status	Shows if LLDP is enabled.

Command History

Release	Modification
7.1	This command was introduced.

Usage Guidelines

The **via** field shows LLDP if it is active, and shows Unknown if LLDP is disabled or not functional.

Examples

The following is sample output from the **show lldp neighbors** command:

```
> show lldp neighbors

--------------------------------------------------------------------------------
LLDP neighbors:
--------------------------------------------------------------------------------
Interface: lldp-Eth1_6, via: LLDP, RID: 1, Time: 0 day, 00:00:18
  Chassis:
    ChassisID: mac 8c:60:4f:58:c1:ac
    SysName: ruintpo
    SysDescr: Cisco Nexus Operating System (NX OS) Software 7.0(1)N1(1)
 TAC support: http://www.cisco.com /tac
 Copyright (c) 2002-2014, Cisco Systems, Inc. All rights reserved.
    MgmtIP: 10.225.126.91
    Capability: Bridge, on
  Port:
    PortID: local Eth1/37
    PortDescr: Ethernet1/37
    TTL: 30
--------------------------------------------------------------------------------
```

The following is sample output from the **show lldp statistics** command:

```
> show lldp statistics interface Ethernet 1/6
--------------------------------------------------------------------------------
LLDP statistics:
--------------------------------------------------------------------------------
Interface: lldp-Eth1_6
  Transmitted: 115
  Received: 116
  Discarded: 0
```

```
  Unrecognized: 0
  Ageout: 0
  Inserted: 0
  Deleted: 0
------------------------------------------------------------------------------
```

The following is sample output from the **show lldp status** command:

```
> show lldp status interface Ethernet 1/6
------------------------------------------------------------------------------
LLDP interfaces:
------------------------------------------------------------------------------
Interface: lldp-Eth1_6, via: unknown, Time: 18795 days, 05:38:39
  Chassis:
    ChassisID: mac 42:8f:14:a8:2f:c5
    SysName: firepower
    SysDescr: Cisco Firepower 1150 Threat Defense 7.1.0 1558
    MgmtIP: 127.128.254.1
    MgmtIP: fd00:0:0:1::3
    Capability: Bridge, on
    Capability: Router, off
    Capability: Wlan , off
    Capability: Station, off
  Port:
    PortID: mac 34:12:78:56:01:03
    PortDescr: Ethernet1/6
    TTL: 120
------------------------------------------------------------------------------
```

Related Commands

Command	Description
show interface	Shows interface statistics.

show local-host

To display the network states of local hosts, use the **show local-host** command.

show local-host [*hostname* | *ip_address*] [**detail**] [**all**] [**brief**] [**connection** {**sctp** | **tcp** | **udp** | **embryonic**} *start*[*-end*]] [**zone**]

Syntax Description

all	(Deprecated) Includes local hosts connecting to and from the device.
brief	(Optional) Displays brief information on local hosts.
connection {**sctp** | **tcp** | **udp** | **embryonic**} *start*[*-end*]	(Deprecated) Applies filters based on the number and type of connections: embryonic, TCP, UDP, or SCTP. The start number indicates the minimum number of connections of that type. Include an -end number to specify a range, such as 10-100. These filters can be used individually or jointly.
detail	(Optional) Displays the detailed network states of local host information, including more information about active xlates and network connections.
hostname | *ip_address*	(Optional) Specifies the local host name or IPv4/IPv6 address.
zone	(Optional) Specifies local hosts per zone or inline set.

Command History

Release	Modification
6.1	This command was introduced.
7.0	The following keywords were deprecated: **all**, **connection**.

Usage Guidelines

The **show local-host** command lets you display the network states of local hosts. A local-host is created for any host that forwards traffic to, or through, the threat defense device.

For systems running 7.0 and later, consider using the **show conn address** command instead of this one.

This command lets you show the translation and connection slots for the local hosts. Translation information includes any PAT port blocks allocated to the host.

This command also displays the connection limit values. If a connection limit is not set, the value displays as 0 and the limit is not applied.

In the event of a SYN attack (with TCP intercept configured), the **show local-host** command output includes the number of intercepted connections in the usage count. This field typically displays only full open connections.

In the **show local-host** command output, the **TCP embryonic count to host counter** is used when a maximum embryonic limit (TCP intercept watermark) is configured for a host using a static connection. This counter shows the total embryonic connections to the host from other hosts. If this total exceeds the maximum configured limit, TCP intercept is applied to new connections to the host.

Examples

The following is sample output from the **show local-host** command:

```
> show local-host

Interface mgmt: 2 active, 2 maximum active, 0 denied
local host: <10.24.250.191>,
    SCTP flow count/limit = 0/unlimited
    TCP flow count/limit = 1/unlimited
    TCP embryonic count to host = 0
    TCP intercept watermark = unlimited
    UDP flow count/limit = 0/unlimited
local host: <10.44.64.65>,
    SCTP flow count/limit = 0/unlimited
    TCP flow count/limit = 1/unlimited
    TCP embryonic count to host = 1
    TCP intercept watermark = unlimited
    UDP flow count/limit = 5/unlimited
Interface inside: 0 active, 0 maximum active, 0 denied
Interface outside: 0 active, 0 maximum active, 0 denied
Interface any: 0 active, 0 maximum active, 0 denied
```

The following examples show the network states of local hosts:

```
> show local-host all

Interface outside: 1 active, 2 maximum active, 0 denied
local host: <11.0.0.4>,
SCTP flow count/limit = 0/unlimited
TCP flow count/limit = 0/unlimited
TCP embryonic count to host = 0
TCP intercept watermark = unlimited
UDP flow count/limit = 0/unlimited
Conn:
105 out 11.0.0.4 in 11.0.0.3 idle 0:01:42 bytes 4464
105 out 11.0.0.4 in 11.0.0.3 idle 0:01:44 bytes 4464
Interface inside: 1 active, 2 maximum active, 0 denied
local host: <17.3.8.2>,
SCTP flow count/limit = 0/unlimited
TCP flow count/limit = 0/unlimited
TCP embryonic count to host = 0
TCP intercept watermark = unlimited
UDP flow count/limit = 0/unlimited
Conn:
105 out 17.3.8.2 in 17.3.8.1 idle 0:01:42 bytes 4464
105 out 17.3.8.2 in 17.3.8.1 idle 0:01:44 bytes 4464
Interface NP Identity Ifc: 2 active, 4 maximum active, 0 denied
local host: <11.0.0.3>,
SCTP flow count/limit = 0/unlimited
TCP flow count/limit = 0/unlimited
TCP embryonic count to host = 0
TCP intercept watermark = unlimited
UDP flow count/limit = 0/unlimited
Conn:
105 out 11.0.0.4 in 11.0.0.3 idle 0:01:44 bytes 4464
105 out 11.0.0.4 in 11.0.0.3 idle 0:01:42 bytes 4464
local host: <17.3.8.1>,
SCTP flow count/limit = 0/unlimited
TCP flow count/limit = 0/unlimited
TCP embryonic count to host = 0
TCP intercept watermark = unlimited
UDP flow count/limit = 0/unlimited
Conn:
105 out 17.3.8.2 in 17.3.8.1 idle 0:01:44 bytes 4464
```

```
105 out 17.3.8.2 in 17.3.8.1 idle 0:01:42 bytes 4464
```

The following example shows information about a specific host, followed by detailed information for that host.

```
> show local-host 10.1.1.91
Interface third: 0 active, 0 maximum active, 0 denied
Interface inside: 1 active, 1 maximum active, 0 denied
local host: <10.1.1.91>,
SCTP flow count/limit = 0/unlimited
TCP flow count/limit = 1/unlimited
TCP embryonic count to (from) host = 0 (0)
TCP intercept watermark = unlimited
UDP flow count/limit = 0/unlimited

Xlate:
PAT Global 192.150.49.1(1024) Local 10.1.1.91(4984)

Conn:
TCP out 192.150.49.10:21 in 10.1.1.91:4984 idle 0:00:07 bytes 75 flags UI Interface
outside: 1 active, 1 maximum active, 0 denied

> show local-host 10.1.1.91 detail
Interface third: 0 active, 0 maximum active, 0 denied
Interface inside: 1 active, 1 maximum active, 0 denied
local host: <10.1.1.91>,
SCTP flow count/limit = 0/unlimited
TCP flow count/limit = 1/unlimited
TCP embryonic count to (from) host = 0 (0)
TCP intercept watermark = unlimited
UDP flow count/limit = 0/unlimited

Xlate:
TCP PAT from inside:10.1.1.91/4984 to outside:192.150.49.1/1024 flags ri

Conn:
TCP outside:192.150.49.10/21 inside:10.1.1.91/4984 flags UI Interface outside: 1 active,
1 maximum active, 0 denied
```

The following example shows all hosts who have at least four UDP connections and have between one to 10 TCP connections at the same time:

```
> show local-host connection udp 4 tcp 1-10
Interface mng: 0 active, 3 maximum active, 0 denied
Interface INSIDE: 4 active, 5 maximum active, 0 denied
local host: <10.1.1.11>,
       TCP flow count/limit = 1/unlimited TCP embryonic count to host = 0 TCP intercept
       watermark = unlimited UDP flow count/limit = 4/unlimited
Xlate:
       Global 192.168.1.24 Local 10.1.1.11 Conn: UDP out 192.168.1.10:80 in
       10.1.1.11:1730 idle 0:00:21 bytes 0 flags - UDP out 192.168.1.10:80 in
       10.1.1.11:1729 idle 0:00:22 bytes 0 flags - UDP out 192.168.1.10:80 in
       10.1.1.11:1728 idle 0:00:23 bytes 0 flags - UDP out 192.168.1.10:80 in
       10.1.1.11:1727 idle 0:00:24 bytes 0 flags - TCP out 192.168.1.10:22 in
       10.1.1.11:27337 idle 0:01:55 bytes 2641 flags UIO Interface OUTSIDE: 3 active, 5
       maximum active, 0 denied
```

Related Commands

Command	Description
clear local-host	Releases network connections from local hosts displayed by the **show local-host** command.

show log-events-to-ramdisk

To display the status of logging connection events to RAM disk, use the **show log-events-to-ramdisk** command.

show log-events-to-ramdisk

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

This command shows whether you are logging connection events to RAM disk or to the Solid State Drive (SSD). RAM disk logging is not supported on all hardware models. You configure RAM disk logging with the **configure log-events-to-ramdisk** command.

Examples

The following example shows that logging to RAM disk is not supported on this hardware model.

```
> show log-events-to-ramdisk
This command is not available on this platform.
```

Related Commands

Command	Description
configure log-events-to-ramdisk	Enables or disables logging connection events to RAM disk.

show logging

To show the logs in the buffer or other logging settings, use the **show logging** command.

show logging [**message** [*syslog_id* | **all**] | **asdm** | **flow-export-syslogs** | **queue** | **setting** | **unified-client** [**statistics**]]

Syntax Description

all	(Optional) Displays all syslog message IDs, along with whether they are enabled or disabled.
asdm	(Optional) This keyword does not work for device manager. It relates to ASDM, which configures ASA Software devices.
flow-export-syslogs	(Optional. Display all of the syslog messages whose information is also captured by NetFlow.
message [*syslog_id* | **all**]	(Optional) If you do not specify a syslog ID or all, this keyword displays messages that are at a non-default level. You can also display messages by ID, or see information on all syslog messages.
queue	(Optional) Displays the syslog message queue.
setting	(Optional) Displays the logging setting, without displaying the logging buffer.
syslog_id	(Optional) Specifies a message number to display.
unified-client [**statistics**]	Shows detailed statistics about the status of the syslog client including the loggerD service status, syslog client registration information, loggerD heartbeat details, and syslog client control/data and error statistics,

Command History

Release	Modification
6.1	This command was introduced.
6.3	The **unified-client** [**statistics**] keyword was added.

Usage Guidelines

If you enable logging to the internal buffer, the **show logging** command without any keywords shows the current message buffer and the current settings.

The **show logging queue** command allows you to display the following:

- Number of messages that are in the queue
- Highest number of messages recorded that are in the queue
- Number of messages that are discarded because block memory was not available to process them
- Separate queues for traps and other syslog messages

Note Zero is an acceptable number for the configured queue size and represents the maximum queue size allowed. The output for the **show logging queue** command will display the actual queue size if the configured queue size is zero.

The **show logging flow-export-syslogs** command shows whether the following syslogs are enabled or disabled. When using Netflow, you have the option of disabling these syslogs because they are redundant.

Syslog Message	Description
106015	A TCP flow was denied because the first packet was not a SYN packet.
106023	A flow that is denied by an ingress ACL or an egress ACL that is attached to an interface.
106100	A flow that is permitted or denied by an ACL.
302013 and 302014	A TCP connection and deletion.
302015 and 302016	A UDP connection and deletion.
302017 and 302018	A GRE connection and deletion.
302020 and 302021	An ICMP connection and deletion.
313001	An ICMP packet to the threat defense device was denied.
313008	An ICMPv6 packet to the threat defense device was denied.
710003	An attempt to connect to the threat defense was denied.

Examples

The following is sample output from the **show logging** command:

```
> show logging
Syslog logging: enabled
    Facility: 20
    Timestamp logging: enabled
    Standby logging:disabled
    Debug-trace logging: disabled
    Console logging: level informational, 3962 messages logged
    Monitor logging: disabled
    Buffer logging: disabled
    Trap logging: level informational, facility 20, 20549 messages logged
        Logging to inside 10.2.5.3 tcp/50001 connected
    Permit-hostdown state
    History logging: disabled
    Device ID: disabled
    Mail logging: disabled
    ASDM logging: disabled
```

Note The possible values for Syslog Logging are enabled, disabled, disabled-blocking, and disabled-not blocking.

The following is sample output from the **show logging**command with a secure syslog server configured:

```
> show logging
Syslog logging: disabled
    Facility:
    Timestamp logging: disabled
    Deny Conn when Queue Full: disabled
    Console logging: level debugging, 135 messages logged
    Monitor logging: disabled
    Buffer logging: disabled
    Trap logging: list show _syslog, facility, 20, 21 messages logged
        Logging to inside 10.0.0.1 tcp/1500 SECURE
    History logging: disabled
    Device ID: disabled
    Mail logging: disabled
    ASDM logging disabled
```

The following is sample output from the **show logging queue** command:

```
> show logging queue
Logging Queue length limit: 512 msg(s)
0 msg(s) discarded due to queue overflow
0 msg(s) discarded due to memory allocation failure
Current 0 msgs on queue, 0 msgs most on queue
```

The following is sample output from the **show logging message all** command:

```
> show logging message all
syslog 111111: default-level alerts (enabled)
syslog 101001: default-level alerts (enabled)
syslog 101002: default-level alerts (enabled)
syslog 101003: default-level alerts (enabled)
syslog 101004: default-level alerts (enabled)
syslog 101005: default-level alerts (enabled)
syslog 102001: default-level alerts (enabled)
syslog 103001: default-level alerts (enabled)
syslog 103002: default-level alerts (enabled)
syslog 103003: default-level alerts (enabled)
syslog 103004: default-level alerts (enabled)
syslog 103005: default-level alerts (enabled)
syslog 103011: default-level alerts (enabled)
syslog 103012: default-level informational (enabled)
```

The following is sample output from the **show logging unified-client** command:

```
> show logging unified-client
Log client details:
 Name : Lina
 Id : 1331
 Init time : Fri Sep 7 07:20:14 2018
 Status : Registered
```

The following is sample output from the **show logging unified-client statistics** command:

```
> show logging unified-client statistics
Log client details:
  Name                                          : Lina
  Id                                            : 1331
  Init time                                     : Fri Sep  7 07:20:14 2018
  Status                                        : Registered

Loggerd service up/down statistics:
  Service status                                : Up
  Instance-id                                   : 4602
  Last service down time                        : Wed Sep 12 05:17:43 2018

Log client register/unregister statistics:
  Total register messages Tx                    : 1222
  Total unregister messages Tx                  : 0
  Last register message Tx time                 : Wed Sep 12 05:40:16 2018
  Total register-ack messages Rx                : 39
  Last register-ack Rx time                     : Wed Sep 12 05:40:17 2018
  Total configuration sent messages Tx          : 14
  Number of configuration pushes                : 38

Heartbeat statistics:
  Last heartbeat Tx time                        : Wed Sep 12 06:38:33 2018
  Last Tx seqnum                                : 10019
  Total heartbeat Tx                            : 9981

Loggerd heartbeat statistics:
  Last heartbeat Rx time                        : Wed Sep 12 06:38:36 2018
  Last hearbeat Rx seqnum                       : 701
  Total heartbeat Rx                            : 5977
  Miss count                                    : 1

Log client data messages details:
  Syslogs Tx for ngfw-management                : 6554
  Syslogs Rx for data ports                     : 0
  Syslogs Tx drops for ngfw-management          : 0

Log client Control/Data channel statistics:
  Total control messages Tx                     : 11757
  Total service messages Rx                     : 98
  Total notify messages Rx                      : 6020
  Total data messages Rx                        : 0

Log-client error statistics:
  Register messages Tx                          : 2373
  Register-ack messages Rx                      : 5921
  Configuration push Tx                         : 1
  Heartbeat Tx                                  : 0
  Control channel Rx                            : 0
  Data channel Rx                               : 0
  Syslogs Rx for data ports                     : 0
```

show mac-address-table

To show the MAC address table, use the **show mac-address-table** command.

show mac-address-table [*interface_name* | **count** | **static**]

Syntax Description

count	(Optional) Lists the total number of dynamic and static entries.
interface_name	(Optional) Identifies the interface name for which you want to view MAC address table entries.
static	(Optional) Lists only static entries.

Command Default

If you do not specify an interface, all interface MAC address entries are shown.

Command History

Release	Modification
6.1	This command was added.
6.2	We added support in routed firewall mode when using Integrated Routing and Bridging.

Examples

The following is sample output from the **show mac-address-table** command:

```
> show mac-address-table
interface    mac address     type      Time Left
-----------------------------------------------------
outside      0009.7cbe.2100  static    -
inside       0010.7cbe.6101  static    -
inside       0009.7cbe.5101  dynamic   10
```

The following is sample output from the **show mac-address-table count** command:

```
> show mac-address-table count
Static     mac-address bridges (curr/max): 0/65535
Dynamic    mac-address bridges (curr/max): 103/65535
```

show mac-learn

To show whether MAC learning is enabled or disabled for each interface, use the **show mac-learn** command.

show mac-learn

Command History

Release	Modification
6.1	This command was added.
6.2	We added support in routed firewall mode when using Integrated Routing and Bridging.

Usage Guidelines

By default, each interface automatically learns the MAC addresses of entering traffic, and the system adds corresponding entries to the MAC address table. You can disable MAC learning per interface.

Examples

The following is sample output from the **show mac-learn** command.

```
> show mac-learn
no mac-learn flood
interface                          mac learn
-----------------------------------------
 outside                             enabled
 inside1_2                           enabled
 inside1_3                           enabled
 inside1_4                           enabled
 inside1_5                           enabled
 inside1_6                           enabled
 inside1_7                           enabled
 inside1_8                           enabled
 diagnostic                          enabled
 inside                              enabled
```

show managers

To show the current manager that is managing the device configuration, use the **show managers** command.

show managers

Command History

Release	Modification
6.1	This command was introduced.
7.2	Added support for multiple managers. The output now includes the management center display name, identifier, and the management type, either Configuration or Analytics.

Usage Guidelines

Use the **show managers** command to determine which application is defined for managing the device configuration. You can then log into the manager using a web browser.

When you configure a remote manager, management center, for the device using the **configure manager add** command, the output shows the host address and registration status. The registration key and NAT ID are only displayed if registration is pending. If a device is registered to a high availability pair, information about both managing Management Centers is displayed. If a device is configured as a secondary device in a stacked configuration, information about both the managing Management Center and the primary device is displayed.

Examples

The following example shows a completed registration to a management center remote manager.

```
> show managers
Type                    : Manager
Host                    : 10.10.1.4
Display name            : 10.10.1.4
Identifier              : f7ffad78-bf16-11ec-a737-baa2f76ef602
Registration            : Completed
Management type         : Configuration
```

The following example shows that the local manager, device manager, is enabled.

```
> show managers
Managed locally.
```

The following example shows that no manager is currently configured. You must use the configure manager add or configure manager local to enable one before you can configure the device.

```
> show managers
No managers configured.
```

The following example shows three managers: one is pending and not currently in use; one is the main configuration manager (CDO); and one is an on-prem analytics-only manager.

```
> show managers
```

```
Type                    : Manager
Host                    : 1.2.3.4
Display name            : 1.2.3.4
Identifier              : 1.2.3.4
Registration            : Pending

Type                    : Manager
Host                    : 10.10.1.4
Display name            : 10.10.1.4
Identifier              : f7ffad78-bf16-11ec-a737-baa2f76ef602
Registration            : Completed
Management type         : Configuration

Type                    : Manager
Host                    : 10.10.2.7
Display name            : 10.10.2.7
Identifier              : 6d3df56e-bf16-11ec-972b-b07a16ffdd03
Registration            : Completed
Management type         : Analytics
```

Related Commands

Command	Description
configure manager add	Adds a remote manager, management center.
configure manager delete	Deletes the current manager and enters No Manager Mode.
configure manager local	Enables the local manager, device manager.

show memory

To display a summary of the maximum physical memory and current free memory available to the operating system, use the **show memory** command.

show memory [**api** | **app-cache** | **binsize** *size* | **caller-address** | **detail** | **region** | **system** | **top-usage** [*num*]]

Syntax Description

api	(Optional) Displays the malloc stack APIs that are registered in the system. If any of the memory debugging features are turned on (that is, delay-free-poisoner, memory logger, memory tracker, or memory profiler), their APIs appear in the output.
app-cache	(Optional) Displays memory usage by application.
binsize *size*	(Optional) Displays summary information about the chunks (memory blocks) allocated for a specific bin size. The bin size is from the “fragment size” column of the **show memory detail** command output.
caller-address	Display information related to the **memory caller-address** configuration.
detail	(Optional) Displays a detailed view of free and allocated system memory.
region	Displays process maps.
system	Displays the total memory, the memory in use, and the available memory for the device.
top-usage [*num*]	Displays the top number of allocated fragment sizes from the **show memory detail** command. You can optionally specify the number of bin sizes to list, from 1-64. The default is 10.

Command History

Release	Modification
6.1	This command was introduced.
6.2.2	Output was changed for **show memory** and **show memory detail**.

Usage Guidelines

The **show memory** command lets you display a summary of the maximum physical memory and current free memory available to the operating system. Memory is allocated as needed.

You can also display the information from the **show memory** command using SNMP.

You can use the **show memory detail** output with the **show memory binsize** command to debug memory leaks.

The **show memory detail** command output can be broken down into three sections: Summary, DMA Memory, and HEAP Memory. The summary displays how the total memory is allocated. Memory that is not tied to DMA or reserved is considered the HEAP. The Free Memory value is the unused memory in the HEAP. The Allocated memory in use value is how much of the HEAP has been allocated. The breakdown of HEAP

allocation is displayed later in the output. Reserved memory and DMA Reserved memory are used by different system processes and primarily VPN services.

The Free memory is divided in to two parts: Free memory heap and Free memory system. Free memory heap is the amount of free memory in the glibc heap. As the glibc heap grows and shrinks on demand, the amount of free heap memory does not indicate the total memory left in the system. Free memory system represents the amount of free memory available to the ASA.

Reserved memory (DMA) is the amount of memory reserved for the DMA pools. Memory overhead is the glibc overhead and process overhead of various running processes.

Values displayed in the allocated memory statistics total (bytes) column do not reflect real values (MEMPOOL_GLOBAL_SHARED POOL STATS) in the **show memory detail** command output.

Note MEMPOOL_GLOBAL_SHARED does not take all the system memory during bootup, but asks the underlying operating system for memory whenever required. Similarly, it returns memory to the system when a significant amount of memory is freed. As a result, the size of MEMPOOL_GLOBAL_SHARED appears to grow and shrink according to demand. A minimal amount of free memory remains in MEMPOOL_GLOBAL_SHARED to speed up allocation.

The output shows that the block of size 49,152 was allocated then returned to the free pool, and another block of size 131,072 was allocated. In this case, you would think that free memory decreased by 131,072-49,152=81,920 bytes, but it actually decreased by 100,000 bytes (see the Free memory line).

```
> show memory detail
MEMPOOL_GLOBAL_SHARED POOL STATS:            MEMPOOL_GLOBAL_SHARED POOL STATS:
Non-mmapped bytes allocated =   1862270976   Non-mmapped bytes allocated =   1862270976
Number of free chunks       =           99   Number of free chunks       =          100
Number of mmapped regions   =            0   Number of mmapped regions   =            0
Mmapped bytes allocated     =            0   Mmapped bytes allocated     =            0
Max memory footprint        =   1862270976   Max memory footprint        =   1862270976
Keepcost                    =   1762019304   Keepcost                    =   1761869256
Max contiguous free mem     =   1762019304   Max contiguous free mem     =   1761869256
Allocated memory in use     =    100133944   Allocated memory in use     =    100233944
Free memory                 =   1762137032   Free memory                 =   1762037032
----- fragmented memory statistics -----     ----- fragmented memory statistics -----
 fragment size       count          total     fragment size       count          total
    (bytes)                       (bytes)       (bytes)                       (bytes)
----------------  ----------  -------------- ----------------  ----------  --------------
        32768             1          33176          32768             1          33176
                                                    49152             1          50048
   1762019304             1     1762019304*     1761869256             1     1761869256*
----- allocated memory statistics -----      ----- allocated memory statistics -----
 fragment size       count          total     fragment size       count          total
    (bytes)                       (bytes)       (bytes)                       (bytes)
----------------  ----------  -------------- ----------------  ----------  --------------
        49152            10         491520          49152             9         442368
        65536           125        8192000          65536           125        8192000
        98304             3         294912          98304             3         294912
       131072            18        2359296         131072            19        2490368
```

The following output confirms that a block of size 150,000 was allocated, instead of 131,072:

```
> show memory binsize 131072
MEMPOOL_DMA pool bin stats:
MEMPOOL_GLOBAL_SHARED pool bin stats:
```

```
pc = 0x8eda524, size = 150000   , count = 1
pc = 0x8f08054, size = 163904   , count = 1
pc = 0x846e477, size = 139264   , count = 1
pc = 0x8068691, size = 393216   , count = 3
pc = 0x8eea09b, size = 131072   , count = 1
pc = 0x88ca830, size = 141212   , count = 1
pc = 0x9589e93, size = 593580   , count = 4
pc = 0x9589bd2, size = 616004   , count = 4
pc = 0x8f2e060, size = 327808   , count = 2
pc = 0x8068284, size = 182000   , count = 1

0x8eda524 <logger_buffer_init_int+148 at syslog/main.c:403>
```

The approximate number of total bytes shown in the **show memory detail** command output is by design. There are two reasons for this:

- For each fragment size, if you had to get the sum of all fragments, a performance impact would occur because there can be very large number of allocations for a single fragment size and to get the accurate value, you need to walk over thousands of chunks.
- For each binsize, you need to walk through the doubly linked list of allocations and there could be many allocations. In this case, you cannot hog the CPU for an extended period and would need to suspend allocations periodically. After you resume allocations, other processes may have allocated or deallocated memory and memory states may have changed. As a result, the total bytes column gives an approximate value instead of the real value.

Examples

The following is sample output from the **show memory** command:

```
> show memory
Free memory:        2986716635 bytes (64%)
Used memory:        1646723072 bytes (36%)
-------------     ------------------
Total memory:       4633439707 bytes (100%)

Note: Free memory is the free system memory. Additional memory may
      be available from memory pools internal to the ASA process.
      Use 'show memory detail' to see this information, but use it
      with care since it may cause CPU hogs and packet loss under load.
>
```

The following example shows how to display system-level memory usage.

```
> show memory system
             total       used       free     shared    buffers     cached
Mem:       3982640    3014544     240200          0     159932     567964
-/+ buffers/cache:    3014544     968096
Swap:      3998716     137704    3861012
```

The following is sample output from the **show memory detail** command:

```
> show memory detail

Heap Memory:
   Free Memory:
```

```
      Heapcache Pool:                        3804848 bytes (  0% )
      Global Shared Pool:                   67372768 bytes (  1% )
      System:                             2986716635 bytes ( 64% )
   Used Memory:
      Heapcache Pool:                      308670800 bytes (  7% )
      Global Shared Pool:                       6432 bytes (  0% )
      Reserved (Size of DMA Pool):         499122176 bytes ( 11% )
      Reserved for messaging:                2097152 bytes (  0% )
      System Overhead:                     765648896 bytes ( 17% )
-------------------------------------     ----------------
   Total Memory:                         4633439707 bytes ( 100% )

Warning:  The information reported here is computationally expensive to
          determine, and may result in CPU hogs and performance impact.

------------------------------------------------------------------------
MEMPOOL_MSGLYR POOL STATS:

Non-mmapped bytes allocated =      2097152
Number of free chunks       =            1
Number of mmapped regions   =            0
Mmapped bytes allocated     =            0
Max memory footprint        =      2097152
Keepcost                    =      2092768
Max contiguous free mem     =      2092768
Allocated memory in use     =         4288
Free memory                 =      2092864

----- fragmented memory statistics -----

(...Remaining output truncated...)
```

The following example shows the chunks allocated to bin size 8192.

```
> show memory binsize 8192
MEMPOOL_HEAPCACHE_0 pool bin stats:
pc = 0x7efc3f80e508, size = 773406   , count = 92
pc = 0x7efc3e3c5013, size = 189152   , count = 23
pc = 0x7efc405df64f, size = 287036   , count = 32
pc = 0x7efc3f9ef622, size = 8128     , count = 1
pc = 0x7efc3f4fd5f5, size = 871744   , count = 106
pc = 0x7efc3f4fd8b7, size = 82240    , count = 10
pc = 0x7efc3f18c3e6, size = 20272    , count = 2
pc = 0x7efc3f557139, size = 8192     , count = 1
pc = 0x7efc3e3f1697, size = 8344     , count = 1
pc = 0x7efc3e0506f6, size = 8192     , count = 1
MEMPOOL_DMA pool bin stats:
pc = 0x7efc3e1cca68, size = 10240    , count = 1
MEMPOOL_GLOBAL_SHARED pool bin stats:
```

This following is sample output from the **show memory api** command. It shows that the memory tracker and delayed-free-poisoner memory features are active.

```
> show memory api
Resource Manager (0) ->
Tracking (0) ->
Delayed-free-poisoner (0) ->
Core malloc package (0)
```

The following example shows how to display system-level memory usage.

```
> show memory system
             total       used       free     shared    buffers     cached
Mem:       3982640    3014544     240200          0     159932     567964
-/+ buffers/cache:    3014544     968096
Swap:      3998716     137704    3861012
```

Related Commands

Command	Description
show memory profile	Displays information about the memory usage (profiling) of the threat defense.

show memory all

To display a summary of the maximum physical memory and current free memory available to the operating system of both lina and Snort, use the **show memory all** command.

show memory all

Command History

Release	Modification
7.0	This command was introduced.

Usage Guidelines

The **show memory all** command lets you display a summary of the maximum physical memory and current free memory available to the operating system. Memory is allocated as needed.

```
> show memory all
Data Path:
Free memory:        3161408675 bytes (72%)
Used memory:        1203826208 bytes (28%)
-------------     ------------------
Total memory:       4365234883 bytes (100%)
Inspection Engine:
Free memory:                  0 bytes ( 0%)
Used memory:                  0 bytes ( 0%)
-------------     ------------------
Total memory:                 0 bytes (100%)
System:
Free memory:                  0 bytes ( 0%)
Used memory:                  0 bytes ( 0%)
-------------     ------------------
Total memory:                 0 bytes (100%)
```

show memory delayed-free-poisoner

To display a summary of the **memory delayed-free-poisoner** queue usage, use the **show memory delayed-free-poisoner** command.

show memory delayed-free-poisoner

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **memory delayed-free-poisoner enable** command to enable the feature. Use the **clear memory delayed-free-poisoner** command to clear the queue and statistics.

Examples

This following is sample output from the **show memory delayed-free-poisoner** command:

```
> memory delayed-free-poisoner enable
> show memory delayed-free-poisoner
delayed-free-poisoner settings:
 delayed-free-poisoner threshold 100
 delayed-free-poisoner desired-fragment-size 102400
 delayed-free-poisoner desired-fragment-count 16
 delayed-free-poisoner watchdog-percent 50
delayed-free-poisoner statistics:
                  136064:  current memory in queue
                     500:  current queue length
                       0:  frees dequeued
                     280:  frees not queued for size
                       0:  frees not queued for locking
                       0:  successful validate runs
                       0:  aborted validate runs
                   never:  time of last validate
                       0:  threshold defragment operations
                       0:  size and/or count defragment operations
                       0:  watchdog-aborts
```

show memory logging

To display memory usage logging, use the **show memory logging** command.

show memory logging [**wrap** | **brief** | **include** [*option*]]

Syntax Description

brief	(Optional) Displays abbreviated memory usage logging.
include *option*	(Optional) Includes only the specified fields in the output. You can specify the keywords for the fields in any order, but they always appear in the following order. If you do not include an option, the output is the same as if you had specified **brief** instead of **include**. • **process** • **time** • **operator** (free/malloc/etc.) • **address** • **size** • **callers** The output format is: process=[XXX] time=[XXX] oper=[XXX] address=0xXXXXXXXX size=XX @ XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX Up to four caller addresses appear. The types of operations are listed in the output (Number of...) shown in the example.
wrap	(Optional) Displays memory usage logging wrapped data, which is purged after you enter this command so that duplicate data does not appear and is not saved.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **show memory logging** command to view memory log information. You must first enable this logging using the **memory logging** command.

Examples

The following is sample output from the **show memory logging** command.

```
> memory logging 1024
> show memory logging
Number of free                          203989
Number of calloc                        83703
```

```
Number of malloc                          120286
Number of realloc-new                     0
Number of realloc-free                    0
Number of realloc-null                    0
Number of realloc-same                    0
Number of calloc-fail                     0
Number of malloc-fail                     0
Number of realloc-fail                    0
Total operations 407978
Buffer size: 1024 (73816 x2 bytes)
process=[cli_xml_server] time=[19:23:42.030] oper=[malloc] addr=0x00007efc358373c0 size=72

@ 0x00007efc3f8e9404 0x00007efc3f80e508 0x00007efc3f4d3cea 0x00007efc3e037f0c
process=[cli_xml_server] time=[19:23:42.030] oper=[free] addr=0x00007efc358373c0 size=72
@ 0x00007efc3f80e9c0 0x00007efc3f4d3fb8 0x00007efc3e037fb0 0x00007efc3f4d537d
(...Remaining output truncated...)
```

The following is sample output from the **show memory logging brief** command.

```
> show memory logging brief
Number of free                            223195
Number of calloc                          91624
Number of malloc                          131572
Number of realloc-new                     0
Number of realloc-free                    0
Number of realloc-null                    0
Number of realloc-same                    0
Number of calloc-fail                     0
Number of malloc-fail                     0
Number of realloc-fail                    0
Total operations 446391
Buffer size: 1024 (73816 x2 bytes)
```

Related Commands

Command	Description
memory logging	Enables memory logging.

show memory profile

To display information about the memory usage (profiling) of the threat defense device, use the **show memory profile** command.

show memory profile [**status** | **peak** [**detail** | **collated**]]

Syntax Description

collated	(Optional) Collates the memory information displayed.
detail	(Optional) Displays detailed memory information.
peak	(Optional) Displays the peak capture buffer rather than the "in use" buffer.
status	(Optional) Displays the current state of memory profiling and the peak capture buffer.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **show memory profile** command to troubleshoot memory usage level and memory leaks. You can still see the profile buffer contents even if profiling has been stopped. Starting profiling clears the buffer automatically.

Note The threat defense device might experience a temporary reduction in performance when memory profiling is enabled.

Examples

The following is sample output from the **show memory profile** command:

```
> show memory profile
Range: start = 0x004018b4, end = 0x004169d0, increment = 00000004
Total = 0
```

The output of the **show memory profile detail** command is divided into six data columns and one header column, at the far left. The address of the memory bucket corresponding to the first data column is given at the header column (the hexadecimal number). The data itself is the number of bytes that is held by the text/code that falls in the bucket address. A period (.) in the data column means no memory is held by the text at this bucket. Other columns in the row correspond to the bucket address that is greater than the increment amount from the previous column. For example, the address bucket of the first data column in the first row is 0x001069e0. The address bucket of the second data column in the first row is 0x001069e4 and so on. Normally the header column address is the next bucket address; that is, the address of the last data column of the previous row plus the increment. All rows without any usage are suppressed. More than one such contiguous row can be suppressed, indicated with three periods at the header column (...).

The following is sample output from the **show memory profile peak detail** command, which shows the peak capture buffer and the number of bytes that is held by the text/code that falls in the corresponding bucket address:

```
> show memory profile peak detail
Range: start = 0x00100020, end = 0x00e006e0, increment = 00000004
Total = 48941152
...
0x001069e0 . 24462 . . . .
...
0x00106d88 . 1865870 . . . .
...
0x0010adf0 . 7788 . . . .
...
0x00113640 . . . . 433152 .
...
0x00116790 2480 . . . . .
(...output truncated...)
```

The following is sample output from the **show memory profile peak collated** command:

```
> show memory profile peak collated
Range: start = 0x00100020, end = 0x00e006e0, increment = 00000004
Total = 48941152
24462 0x001069e4
1865870 0x00106d8c
7788 0x0010adf4
433152 0x00113650
2480 0x00116790
<More>
```

The following is sample output from the **show memory profile peak** command, which shows the peak capture buffer:

```
> show memory profile peak
Range: start = 0x004018b4, end = 0x004169d0, increment = 00000004
Total = 102400
```

The following is sample output from the **show memory profile status** command, which shows the current state of memory profiling and the peak capture buffer:

```
> show memory profile status
InUse profiling: ON
Peak profiling: OFF
Memory used by profile buffers: 11518860 bytes
Profile:
0x00100020-0x00bfc3a8(00000004)
```

Related Commands

Command	Description
memory profile enable	Enables the monitoring of memory usage (memory profiling).
memory profile text	Configures a program text range of memory to profile.
clear memory profile	Clears the memory buffers held by the memory profiling function.

show memory tracking

To display currently allocated memory tracked by the tool, use the **show memory tracking** command.

show memory tracking [**address** | **detail** | **dump** *tracked_address*]

Syntax Description

address	(Optional) Shows memory tracking by address.
detail	(Optional) Shows the internal memory tracking state.
dump *tracked_address*	(Optional) Shows the dump of the specified memory tracking address, 0-4294967295.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **show memory tracking** command to show currently allocated memory tracked by the tool. You must use **memory tracking enable** before you can see this information.

Examples

The following is sample output from the **show memory tracking** command:

```
> show memory tracking
memory tracking by caller:
  bytes-threshold:        0
  allocates-by-threshold: 0
         65406 bytes from    49 allocates by 0x00007efc3f80e508
          3000 bytes from     1 allocates by 0x00007efc3f4e1278
           159 bytes from     1 allocates by 0x00007efc3fe9ee13
            17 bytes from     1 allocates by 0x00007efc3fe9ef4e
```

The following is sample output from the **show memory tracking address** command:

```
> show memory tracking address
memory tracking by caller:
  bytes-threshold:        0
  allocates-by-threshold: 0
         58918 bytes from    49 allocates by 0x00007efc3f80e508
          3000 bytes from     1 allocates by 0x00007efc3f4e1278
           167 bytes from     1 allocates by 0x00007efc3fe9ee13
            17 bytes from     1 allocates by 0x00007efc3fe9ef4e
memory tracking address pool:
            32 byte region @ 0x00007efc358a06e0 allocated by 0x00007efc3f80e508
            96 byte region @ 0x00007efc351d0880 allocated by 0x00007efc3f80e508
           896 byte region @ 0x00007efc35f121c0 allocated by 0x00007efc3f80e508
          8192 byte region @ 0x00007efc35832e20 allocated by 0x00007efc3f80e508
            96 byte region @ 0x00007efc30483910 allocated by 0x00007efc3f80e508
            88 byte region @ 0x00007efc359e3960 allocated by 0x00007efc3f80e508
          1036 byte region @ 0x00007efc35f04680 allocated by 0x00007efc3f80e508
            76 byte region @ 0x00007efc36024890 allocated by 0x00007efc3f80e508
            24 byte region @ 0x00007efc35fd48a0 allocated by 0x00007efc3f80e508
```

```
  32 byte region @ 0x00007efc35f04ad0 allocated by 0x00007efc3f80e508
  34 byte region @ 0x00007efc35e54e00 allocated by 0x00007efc3f80e508
8192 byte region @ 0x00007efc35834e70 allocated by 0x00007efc3f80e508
  40 byte region @ 0x00007efc36005cc0 allocated by 0x00007efc3f80e508
  11 byte region @ 0x00007efc360061e0 allocated by 0x00007efc3f80e508
  76 byte region @ 0x00007efc357a6dd0 allocated by 0x00007efc3f80e508
1024 byte region @ 0x00007efc358574f0 allocated by 0x00007efc3f80e508
  88 byte region @ 0x00007efc365b7ef0 allocated by 0x00007efc3f80e508
  56 byte region @ 0x00007efc365b7f90 allocated by 0x00007efc3f80e508
 168 byte region @ 0x00007efc365b8210 allocated by 0x00007efc3f80e508
 112 byte region @ 0x00007efc365b8300 allocated by 0x00007efc3f80e508
 112 byte region @ 0x00007efc365b83c0 allocated by 0x00007efc3f80e508
  16 byte region @ 0x00007efc365b8560 allocated by 0x00007efc3f80e508
 167 byte region @ 0x00007efc365b85c0 allocated by 0x00007efc3fe9ee13
2048 byte region @ 0x00007efc357a8610 allocated by 0x00007efc3f80e508
  88 byte region @ 0x00007efc35728be0 allocated by 0x00007efc3f80e508
  88 byte region @ 0x00007efc357a8e60 allocated by 0x00007efc3f80e508
4112 byte region @ 0x00007efc35fe90c0 allocated by 0x00007efc3f80e508
  17 byte region @ 0x00007efc365b95a0 allocated by 0x00007efc3fe9ef4e
  72 byte region @ 0x00007efc365b9600 allocated by 0x00007efc3f80e508
  72 byte region @ 0x00007efc365b9690 allocated by 0x00007efc3f80e508
  72 byte region @ 0x00007efc365b9720 allocated by 0x00007efc3f80e508
  40 byte region @ 0x00007efc365b97b0 allocated by 0x00007efc3f80e508
  24 byte region @ 0x00007efc365b9820 allocated by 0x00007efc3f80e508
   2 byte region @ 0x00007efc365b9880 allocated by 0x00007efc3f80e508
  76 byte region @ 0x00007efc35ff9aa0 allocated by 0x00007efc3f80e508
 776 byte region @ 0x00007efc35f19df0 allocated by 0x00007efc3f80e508
 512 byte region @ 0x00007efc3585a0a0 allocated by 0x00007efc3f80e508
 936 byte region @ 0x00007efc357aaea0 allocated by 0x00007efc3f80e508
  24 byte region @ 0x00007efc357ab290 allocated by 0x00007efc3f80e508
 568 byte region @ 0x00007efc3592bc40 allocated by 0x00007efc3f80e508
 512 byte region @ 0x00007efc35e5c8a0 allocated by 0x00007efc3f80e508
  40 byte region @ 0x00007efc35f2cae0 allocated by 0x00007efc3f80e508
1665 byte region @ 0x00007efc359fcda0 allocated by 0x00007efc3f80e508
 168 byte region @ 0x00007efc34fccf60 allocated by 0x00007efc3f80e508
 112 byte region @ 0x00007efc35ffd0e0 allocated by 0x00007efc3f80e508
4112 byte region @ 0x00007efc356bd340 allocated by 0x00007efc3f80e508
8208 byte region @ 0x00007efc3643d3e0 allocated by 0x00007efc3f80e508
 386 byte region @ 0x00007efc359fd470 allocated by 0x00007efc3f80e508
  72 byte region @ 0x00007efc35e4d570 allocated by 0x00007efc3f80e508
8208 byte region @ 0x00007efc359fd840 allocated by 0x00007efc3f80e508
4112 byte region @ 0x00007efc3592ded0 allocated by 0x00007efc3f80e508
3000 byte region @ 0x00007efc357ee5c0 allocated by 0x00007efc3f4e1278
  32 byte region @ 0x00007efc351be6d0 allocated by 0x00007efc3f80e508
  16 byte region @ 0x00007efc359de790 allocated by 0x00007efc3f80e508
1036 byte region @ 0x00007efc3524f080 allocated by 0x00007efc3f80e508
 512 byte region @ 0x00007efc357ff290 allocated by 0x00007efc3f80e508
 360 byte region @ 0x00007efc357ef360 allocated by 0x00007efc3f80e508
  24 byte region @ 0x00007efc357ff4e0 allocated by 0x00007efc3f80e508
```

Related Commands

Command	Description
clear memory tracking	Clears all currently collected information.
memory tracking	Enables memory tracking.

show memory webvpn

To generate memory usage statistics for WebVPN, use the **show memory webvpn** command.

show memory webvpn [**allobjects** | **blocks** | **dumpstate** *filename* | **pools** | **usedobjects**]
show memory webvpn profile [**clear** | **dump** *filename* | **start** | **stop**]

Syntax Description

allobjects	Displays WebVPN memory consumption details for pools, blocks, and all used and freed objects.
blocks	Displays WebVPN memory consumption details for memory blocks.
clear	Clears the WebVPN memory profile.
dump *filename*	Puts WebVPN memory profile into the specified file. The file name should include the location, which can be disk0:, disk1:, flash:, ftp:, tftp:.
dumpstate *filename*	Puts WebVPN memory state into the specified file. The file name should include the location, which can be disk0:, disk1:, flash:, ftp:, tftp:.
pools	Shows WebVPN memory consumption details for memory pools.
profile	Obtains the WebVPN memory profile and places it in a file.
start	Starts gathering the WebVPN memory profile.
stop	Stops getting the WebVPN memory profile.
usedobjects	Displays WebVPN memory consumption details for used objects.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show memory webvpn allobjects** command:

```
> show memory webvpn allobjects
Arena 0x36b14f8 of 4094744 bytes (61 blocks of size 66048), maximum 134195200
130100456 free bytes (97%; 1969 blocks, zone 0)
Arena is dynamically allocated, not contiguous
Features: GroupMgmt: SET, MemDebugLog: unset
Pool 0xd719a78 ("cp_entries" => "pool for class cpool entries") (next 0xd6d91d8)
Size: 66040 (1% of current, 0% of limit)
Object frame size: 32
Load related limits: 70/50/30
Callbacks: !init/!prep/!f2ca/!dstr/!dump
Blocks in use:
Block 0xd719ac0..0xd729cb8 (size 66040), pool "cp_entries"
Watermarks { 0xd7098f8 <= 0xd70bb60 <= 0xd719a60 } = 57088 ready
Block size 66040 not equal to arena block 66048 (realigned-to-8)
```

```
Used objects: 0
Top allocated count: 275
Objects dump:
0. Object 0xd70bb50: FREED (by "jvclass_pool_free")
```

show mfib

To display information from the Multicast Forwarding Information Base, use the **show mfib** command.

show mfib [*source_or_group* [*group*]] [**cluster** | **count** | **verbose**]
show mfib [**active** [*kbps*] | **cluster-stats** | **interface** | **status** | **summary**]
show mfib reserved [**active** [*kbps*] | **cluster** | **count** | **verbose**]

Syntax Description

[**active** [*kbps*]	(Optional) Displays active multicast sources. You can specify a kilobit per second limit the display to multicast streams that are greater-than or equal to this value. The default is 4, the range is 0-4294967295.
cluster	(Optional) Displays the MFIB epoch number and the current timer value. You cannot specify **cluster** if you specify both a source and group.
cluster-stats	(Optional) Displays MFIB cluster synchronization statistics.
count	(Optional) Displays MFIB route and packet count data. This command displays packet drop statistics.
interface	(Optional) Displays packet statistics for interfaces that are related to the MFIB process.
reserved	(Optional) Displays MFIB entries for reserved groups, in the range 224.0.0.0 through 224.0.0.225.
source_or_group [*group*]	(Optional) The source or group IPv4, IPv6, or name. If you specify both, specify the source first. The source address is a unicast address.
status	(Optional) Displays the general MFIB configuration and operational status.
summary	(Optional) Displays summary information about the number of MFIB entries and interfaces.
verbose	(Optional) Displays detail information about the forwarding entries and interfaces

Command Default

Without the optional arguments, information for all groups is shown.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show mfib** command:

```
> show mfib 224.0.2.39
Entry Flags: C - Directly Connected, S - Signal, IA - Inherit A flag,
             AR - Activity Required, D - Drop
Forwarding counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
```

```
Other counts: Total/RPF failed/Other drops
Interface flags: A - Accept, F - Forward, NS - Negate Signalling
             IC - Internal Copy, NP - Not platform switched
             SP - Signal Present
Interface Counts: FS Pkt Count/PS Pkt Count
(*,224.0.1.39) Flags: S K
  Forwarding: 0/0/0/0, Other: 0/0/0
```

The following is sample output from the **show mfib verbose** command:

```
> show mfib verbose
Entry Flags: C - Directly Connected, S - Signal, IA - Inherit A flag,
             AR - Activity Required, D - Drop
Forwarding counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts: Total/RPF failed/Other drops
Interface flags: A - Accept, F - Forward, NS - Negate Signalling
             IC - Internal Copy, NP - Not platform switched
             SP - Signal Present
Interface Counts: FS Pkt Count/PS Pkt Count
(*,224.0.1.39) Flags: S K
  Forwarding: 0/0/0/0, Other: 0/0/0
(*,224.0.1.40) Flags: S K
  Forwarding: 0/0/0/0, Other: 0/0/0
(*,224.0.0.0/8) Flags: K
  Forwarding: 0/0/0/0, Other: 0/0/0
```

The following sample output from the **show mfib count** command:

```
> show mfib count
MFIB global counters are :
* Packets [no input idb] : 0
* Packets [failed route lookup] : 0
* Packets [Failed idb lookup] : 0
* Packets [Mcast disabled on input I/F] : 0
```

The following is sample output from the **show mfib active** command. The output displays either positive or negative numbers for the rate PPS. The command displays negative numbers when RPF packets fail or when the router observes RPF packets with an interfaces out (OIF) list. This type of activity may indicate a multicast routing problem.

```
> show mfib active
Active IP Multicast Sources - sending >= 4 kbps

Group: 224.2.127.254, (sdr.cisco.com)
   Source: 192.168.28.69 (mbone.ipd.anl.gov)
     Rate: 1 pps/4 kbps(1sec), 4 kbps(last 1 secs), 4 kbps(life avg)

Group: 224.2.201.241, ACM 97
   Source: 192.168.52.160 (webcast3-e1.acm97.interop.net)
     Rate: 9 pps/93 kbps(1sec), 145 kbps(last 20 secs), 85 kbps(life avg)

Group: 224.2.207.215, ACM 97
   Source: 192.168.52.160 (webcast3-e1.acm97.interop.net)
     Rate: 3 pps/31 kbps(1sec), 63 kbps(last 19 secs), 65 kbps(life avg)
```

The following example is sample output from the **show mfib interface** command:

```
> show mfib interface
IP Multicast Forwarding (MFIB) status:
    Configuration Status: enabled
    Operational Status: running
MFIB interface       status    CEF-based output
                               [configured,available]
           Ethernet0   up    [        no,       no]
           Ethernet1   up    [        no,       no]
           Ethernet2   up    [        no,       no]
```

The following is sample output from the **show mfib status** command:

```
> show mfib status
IP Multicast Forwarding (MFIB) status:
    Configuration Status: enabled
    Operational Status: running
```

The following is sample output from the **show mfib summary** command:

```
> show mfib summary
IPv6 MFIB summary:

  54     total entries [1 (S,G), 7 (*,G), 46 (*,G/m)]

  17     total MFIB interfaces
```

The following is sample output from the **show mfib reserved** command:

```
> show mfib reserved
Entry Flags: C - Directly Connected, S - Signal, IA - Inherit A flag,
             AR - Activity Required, D - Drop
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts: Total/RPF failed/Other drops
Interface Flags: A - Accept, F - Forward, NS - Negate Signalling
             IC - Internal Copy, NP - Not platform switched
             SP - Signal Present
Interface Counts: FS Pkt Count/PS Pkt Count
(*,224.0.0.0/4) Flags: C K
   Forwarding: 0/0/0/0, Other: 0/0/0
(*,224.0.0.0/24) Flags: K
   Forwarding: 0/0/0/0, Other: 0/0/0
(*,224.0.0.1) Flags:
   Forwarding: 0/0/0/0, Other: 0/0/0
   outside Flags: IC
   dmz Flags: IC
   inside Flags: IC
```

Related Commands

Command	Description
clear mfib counters	Clears MFIB router packet counters.
show mroute active	Displays active multicast streams.
show mroute count	Displays multicast route counters.
show mroute summary	Displays multicast routing table summary information.

show mgcp

To display Media Gateway Control Protocol (MGCP) configuration and session information, use the **show mgcp** command.

show mgcp {commands | **sessions**} [**detail**]

Syntax Description

commands	Lists the number of MGCP commands in the command queue.
detail	(Optional) Lists additional information about each command or session in the output.
sessions	Lists the number of existing MGCP sessions.

Command History

Release	Modification
6.2.1	This command was introduced.

Usage Guidelines

To display MGCP information, you must inspect MGCP traffic. To inspect MGCP traffic, you need to configure a FlexConfig in management center.

Example

The following are examples of the **show mgcp** command options:

```
> show mgcp commands
1 in use, 1 most used, 200 maximum allowed
CRCX, gateway IP: host-pc-2, transaction ID: 2052, idle: 0:00:07

> show mgcp commands detail
1 in use, 1 most used, 200 maximum allowed
CRCX, idle: 0:00:10
       Gateway IP | host-pc-2
       Transaction ID  2052
       Endpoint name | aaln/1
       Call ID | 9876543210abcdef
       Connection ID |
       Media IP | 192.168.5.7
       Media port | 6058

> show mgcp sessions
1 in use, 1 most used
Gateway IP host-pc-2, connection ID 6789af54c9, active 0:00:11

> show mgcp sessions detail
1 in use, 1 most used
Session active 0:00:14
       Gateway IP | host-pc-2
       Call ID | 9876543210abcdef
       Connection ID | 6789af54c9
       Endpoint name | aaln/1
       Media lcl port  6166
       Media rmt IP | 192.168.5.7
```

```
Media rmt port  6058
```

show mini-coredump status

To display the setting of mini-coredump generation, enter the **show mini-coredump status** command.

show mini-coredump status

Command History

Release	Modification
7.0	This command was introduced.

Usage Guidelines

Mini-coredump generation is enabled by default.

Snort 3 process dumps huge core files because of its multi-threaded nature. These dumps take a while to be written onto the hard disk. Until the core is written and a new process is started, Snort's traffic inspection is interrupted. Creating mini-coredumps avoid time delays. Mini-coredumps have essential details of the stack and memory values which aid in debugging.

Example

The following example shows that mini-coredump generation is disabled.

```
> show mini-coredump status
minicoredump feature status : Disabled
```

Related Commands

Command	Description
configure mini-coredump	Enables or disables mini-coredump generation.

show mode

To show the security context mode for the system, use the **show mode** command.

show mode

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The threat defense device supports single context only. Multiple context mode is not supported.

Examples

The following example shows how to display the security context mode.

```
> show mode
Security context mode: single
```

show model

To display the hardware model of the device, use the **show model** command.

show model

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows the device model.

```
> show model
Cisco ASA5516-X Threat Defense
```

Related Commands

Command	Description
show serial-number	Show the device serial number.
show version	Show software and other device version information.

show module

To show information about a module installed on the threat defense device, use the **show module** command in user EXEC mode.

show module [*id* [**details** | **recover** | **log console**]] | **all**]

Syntax Description

all	(Default) Shows information for all modules. This is the default.
details	(Optional) Shows additional information, including remote management configuration for modules.
id	Specifies the module ID. Use show module without parameters to see the available slot numbers, which are typically 0 and 1.
log console	(Optional) Shows log information for the module. This option might not be valid for every module.
recover	(Optional) Shows the settings for recovering the module.

Command Default

By default, information for all modules is shown.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

This command shows information about the modules installed in the threat defense device. The threat defense itself also appears as a module in the display (in slot 0). Whether a device supports additional modules differs by device model.

The output of the **show module details** command varies according to which module is installed.

For models that allow you to configure software modules, the **show module** command lists all possible modules. Status information indicates whether one of them is installed.

Examples

The following sample output is for an ASA 5516-X running threat defense software. For this device, it is normal for slot 1 to be unknown, because threat defense does not support any software modules.

```
> show module

Mod  Card Type                                    Model              Serial No.
---- ---------------------------------------- ------------------ -----------
   0 ASA 5516-X with FirePOWER services, 8GE, AC, ASA5516            JAD19390056I
   1 Unknown                                    N/A                JAD19390056I

Mod  MAC Address Range                 Hw Version   Fw Version   Sw Version
---- --------------------------------- ------------ ------------ ---------------
   0 84b2.61b1.92be to 84b2.61b1.92c6  1.0          1.1.3        97.1(0)60
   1 84b2.61b1.92bd to 84b2.61b1.92bd  N/A          N/A
```

```
Mod  SSM Application Name           Status           SSM Application Version
---- ------------------------------ ---------------- --------------------------
   1 Unknown                        No Image Present Not Applicable

Mod  Status             Data Plane Status     Compatibility
---- ------------------ --------------------- -------------
   0 Up Sys             Not Applicable
   1 Unresponsive       Not Applicable
```

The following table describes each field listed in the output.

Table 2: show module Output Fields

Field	Description
Mod	The module number, 0 or 1.
Card Type	The card type. For the device shown in module 0, the type is the platform model. For slot 1, it would be the extra module, if any.
Model	The model number for this module.
Serial No.	The serial number.
MAC Address Range	The MAC address range for interfaces on this module.
Hw Version	The hardware version.
Fw Version	The firmware version.
Sw Version	The software version. This is not the threat defense version. Instead, it is an ASA software version, which is a component of threat defense software. Use the **show version** command to see the threat defense version.
SSM Application Name	The name of the application running on the security services module.
SSM Application Version	The version of the application running on the security services module.
Status	For the device in module 0, the status is Up Sys. The status of the module in slot 1 can be any of the following: • Initializing—The module is being detected and the control communication is being initialized by the device. • Up—The module has completed initialization by the device. • Unresponsive—The device encountered an error while communicating with this module. • Reloading—The module is reloading. • Shutting Down—The module is shutting down. • Down—The module is shut down. • Recover—The module is attempting to download a recovery image. • No Image Present—The module software has not been installed.

Field	Description
Data Plane Status	The current state of the data plane.
Compatibility	The compatibility of the module relative to the rest of the device.

show monitor-interface

To display information about the interfaces monitored for failover, use the **show monitor-interface** command.

show monitor-interface

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Because an interface can have more than one IPv6 address configured on it, only the link-local address is displayed in the **show monitor-interface** command. If both IPv4 and IPv6 addresses are configured on an interface, both addresses appear in the output. If there is no IPv4 address configured on the interface, the IPv4 address in the output appears as 0.0.0.0. If there is no IPv6 address configured on an interface, the address is simply omitted from the output.

Monitored failover interfaces can have the following status:

- (Waiting) coupled with any other status, such as Unknown (Waiting)—The interface has not yet received a hello packet from the corresponding interface on the peer unit.
- Unknown—Initial status. This status can also mean the status cannot be determined.
- Normal—The interface is receiving traffic. If the status is Normal (Waiting), verify that a standby IP address has been configured for the interface and that there is connectivity between the two interfaces.
- Testing—Hello messages are not heard on the interface for five poll times.
- Link Down—The interface or VLAN is administratively down.
- No Link—The physical link for the interface is down.
- Failed—No traffic is received on the interface, yet traffic is heard on the peer interface.

Examples

The following is sample output from the **show monitor-interface** command:

```
> show monitor-interface
 This host: Primary - Active
  Interface inside (192.168.1.13): Normal (Monitored)
  Interface outside (192.168.2.13): Normal (Monitored)
 Other host: Secondary - Standby Ready
  Interface inside (192.168.1.14): Normal (Monitored)
  Interface outside (192.168.2.14): Normal (Monitored)
```

show mrib client

To display information about the MRIB client connections, use the **show mrib client** command.

show mrib client [**filter**] [**name** *client_name*]

Syntax Description

filter	(Optional) Displays client filter. Used to view information about the MRIB flags that each client owns and the flags in which each clients is interested.
name *client_name*	(Optional) Name of a multicast routing protocol that acts as a client of MRIB, such as PIM or IGMP.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **filter** option is used to display the route and interface level flag changes that various MRIB clients have registered. This command option also shows what flags are owned by the MRIB clients.

Examples

The following sample output from the **show mrib client** command using the **filter** keyword:

```
> show mrib client filter
MFWD:0 (connection id 0)
interest filter:
entry attributes: S C IA D
interface attributes: F A IC NS DP SP
groups:
include 0.0.0.0/0
interfaces:
include All
ownership filter:
groups:
include 0.0.0.0/0
interfaces:
include All
igmp:77964 (connection id 1)
ownership filter:
interface attributes: II ID LI LD
groups:
include 0.0.0.0/0
interfaces:
include All
pim:49287 (connection id 5)
interest filter:
entry attributes: E
interface attributes: SP II ID LI LD
groups:
include 0.0.0.0/0
interfaces:
include All
ownership filter:
entry attributes: L S C IA D
```

```
interface attributes: F A IC NS DP
groups:
include 0.0.0.0/0
interfaces:
include All
```

Related Commands

Command	Description
show mrib route	Displays MRIB table entries.

show mrib route

To display entries in the MRIB table, use the **show mrib route** command.

show mrib route [[[*source* | *****] [*group*[/*prefix-length*]]] | **summary**]

Syntax Description

*****	(Optional) Display shared tree entries.
/*prefix-length*	(Optional) Prefix length of the MRIB route. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
group	(Optional) IP address or name of the group.
source	(Optional) IP address or name of the route source.
summary	Displays a summary of the MRIB table entries.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The MFIB table maintains a subset of entries and flags updated from MRIB. The flags determine the forwarding and signaling behavior according to a set of forwarding rules for multicast packets.

In addition to the list of interfaces and flags, each route entry shows various counters. Byte count is the number of total bytes forwarded. Packet count is the number of packets received for this entry. The **show mfib count** command displays global counters independent of the routes.

Examples

The following is sample output from the **show mrib route** command:

```
> show mrib route
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept, D - Drop
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
LD - Local Disinterest
(*,224.0.0.0/4) RPF nbr: 10.11.1.20 Flags: L C
   Decapstunnel0 Flags: NS

(*,224.0.0.0/24) Flags: D

(*,224.0.1.39) Flags: S

(*,224.0.1.40) Flags: S
   POS0/3/0/0 Flags: II LI

(*,238.1.1.1) RPF nbr: 10.11.1.20 Flags: C
   POS0/3/0/0 Flags: F NS LI
```

```
   Decapstunnel0 Flags: A

(*,239.1.1.1) RPF nbr: 10.11.1.20 Flags: C
   POS0/3/0/0 Flags: F NS
   Decapstunnel0 Flags: A
```

Related Commands

Command	Description
show mfib count	Displays route and packet count data for the MFIB table.

show mroute

To display the IPv4 multicast routing table, use the **show mroute** command.

show mroute [*group* [*source*] | **reserved**] [**active** [*rate*] | **count** | **pruned** | **summary**]

Syntax Description

active *rate*	(Optional) Displays only active multicast sources. Active sources are those sending at the specified *rate* or higher. If the *rate* is not specified, active sources are those sending at a rate of 4 kbps or higher.
count	(Optional) Displays statistics about the group and source, including number of packets, packets per second, average packet size, and bits per second.
group	(Optional) IP address or name of the multicast group as defined in the DNS hosts table.
pruned	(Optional) Displays pruned routes.
reserved	(Optional) Displays reserved groups.
source	(Optional) Source hostname or IP address.
summary	(Optional) Displays a one-line, abbreviated summary of each entry in the multicast routing table.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **show mroute** command displays the contents of the multicast routing table. The device populates the multicast routing table by creating (S,G) and (*,G) entries based on PIM protocol messages, IGMP reports, and traffic. The asterisk (*) refers to all source addresses, the "S" refers to a single source address, and the "G" is the destination multicast group address. In creating (S, G) entries, the software uses the best path to that destination group found in the unicast routing table (through RPF).

To view the **mroute** commands in the running configuration, use the **show running-config mroute** command.

Examples

The following is sample output from the **show mroute** command:

```
> show mroute

Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT
Timers: Uptime/Expires
Interface state: Interface, State

(*, 239.1.1.40), 08:07:24/never, RP 0.0.0.0, flags: DPC
```

```
  Incoming interface: Null
  RPF nbr: 0.0.0.0
  Outgoing interface list:
    inside, Null, 08:05:45/never
    tftp, Null, 08:07:24/never

(*, 239.2.2.1), 08:07:44/never, RP 140.0.0.70, flags: SCJ
  Incoming interface: outside
  RPF nbr: 140.0.0.70
  Outgoing interface list:
    inside, Forward, 08:07:44/never
```

The following fields are shown in the **show mroute** output:

- Flags—Provides information about the entry.
  - D—Dense. Entry is operating in dense mode.
  - S—Sparse. Entry is operating in sparse mode.
  - B—Bidir Group. Indicates that a multicast group is operating in bidirectional mode.
  - s—SSM Group. Indicates that a multicast group is within the SSM range of IP addresses. This flag is reset if the SSM range changes.
  - C—Connected. A member of the multicast group is present on the directly connected interface.
  - L—Local. The device itself is a member of the multicast group. Groups are joined locally by the igmp join-group command (for the configured group).
  - I—Received Source Specific Host Report. Indicates that an (S, G) entry was created by an (S, G) report. This (S, G) report could have been created by IGMP. This flag is set only on the DR.
  - P—Pruned. Route has been pruned. The software keeps this information so that a downstream member can join the source.
  - R—RP-bit set. Indicates that the (S, G) entry is pointing toward the RP.
  - F—Register flag. Indicates that the software is registering for a multicast source.
  - T—SPT-bit set. Indicates that packets have been received on the shortest path source tree.
  - J—Join SPT. For (*, G) entries, indicates that the rate of traffic flowing down the shared tree is exceeding the SPT-Threshold set for the group. (The default SPT-Threshold setting is 0 kbps.) When the J - Join shortest path tree (SPT) flag is set, the next (S, G) packet received down the shared tree triggers an (S, G) join in the direction of the source, thereby causing the device to join the source tree.

    For (S, G) entries, indicates that the entry was created because the SPT-Threshold for the group was exceeded. When the J - Join SPT flag is set for (S, G) entries, the device monitors the traffic rate on the source tree and attempts to switch back to the shared tree for this source if the traffic rate on the source tree falls below the SPT-Threshold of the group for more than 1 minute.

Note The device measures the traffic rate on the shared tree and compares the measured rate to the SPT-Threshold of the group once every second. If the traffic rate exceeds the SPT-Threshold, the J - Join SPT flag is set on the (*, G) entry until the next measurement of the traffic rate. The flag is cleared when the next packet arrives on the shared tree and a new measurement interval is started.

If the default SPT-Threshold value of 0 kbps is used for the group, the J - Join SPT flag is always set on (*, G) entries and is never cleared. When the default SPT-Threshold value is used, the device immediately switches to the shortest path source tree when traffic from a new source is received.

- Timers:Uptime/Expires—Uptime indicates per interface how long (in hours, minutes, and seconds) the entry has been in the IP multicast routing table. Expires indicates per interface how long (in hours, minutes, and seconds) until the entry will be removed from the IP multicast routing table.
- Interface state—Indicates the state of the incoming or outgoing interface.
  - Interface—The interface name listed in the incoming or outgoing interface list.
  - State—Indicates that packets will either be forwarded, pruned, or null on the interface depending on whether there are restrictions due to access lists or a time-to-live (TTL) threshold.
- (*, 239.1.1.40) and (* , 239.2.2.1)—Entries in the IP multicast routing table. The entry consists of the IP address of the source followed by the IP address of the multicast group. An asterisk (*) in place of the source indicates all sources.
- RP—Address of the RP. For routers and access servers operating in sparse mode, this address is always 224.0.0.0.
- Incoming interface—Expected interface for a multicast packet from the source. If the packet is not received on this interface, it is discarded.
- RPF nbr—IP address of the upstream router to the source.
- Outgoing interface list—Interfaces through which packets will be forwarded.

Related Commands

Command	Description
show running-config mroute	Displays configured multicast routes.

show nameif

To view the logical name for an interface, use the **show nameif** command.

show nameif [*physical_interface*[**.***subinterface*] | **zone**]

Syntax Description

physical_interface	(Optional) Identifies the interface ID, such as **gigabitethernet0/1**.
subinterface	(Optional) Identifies an integer between 1 and 4294967293 designating a logical subinterface.
zone	(Optional) Shows the zone and inline set names.

Command Default

If you do not specify an interface, this command displays all interface names.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use this command to show the names assigned to the interfaces. An interface must be named to use it in any configuration setting. It also shows the security level for the interface, which is always 0 for threat defense.

If you add the **zone** keyword, the Zone Name column indicates the inline set or traffic zone to which the interface belongs. Traffic zone is not the same as security zone, so if you do not have passive interfaces or inline sets, the column might be empty even though the interfaces belong to routed or switched security zones. Use the device manager to determine which security zones contain each interface.

Examples

The following is sample output from the **show nameif** command:

```
> show nameif
Interface               Name                    Security
GigabitEthernet1/1      outside                   0
GigabitEthernet1/2      inside1_2                 0
GigabitEthernet1/3      inside1_3                 0
GigabitEthernet1/4      inside1_4                 0
GigabitEthernet1/5      inside1_5                 0
GigabitEthernet1/6      inside1_6                 0
GigabitEthernet1/7      inside1_7                 0
GigabitEthernet1/8      inside1_8                 0
Management1/1           diagnostic                0
BVI1                    inside                    0
```

The following is sample output that shows zone membership. In this example, 2 interfaces are in inline sets, and one interface is in a passive traffic zone.

```
> show nameif zone
Interface               Name                    Zone Name          Security
GigabitEthernet0/0      passive                 passive-security-zone
                                                                     0
GigabitEthernet0/1      in                      is-154               0
```

```
GigabitEthernet0/2      out                    is-154              0
Management0/0           diagnostic                                 0
```

show nat

To display statistics of NAT policies, use the **show nat** command.

show nat [**interface** *name*] [*ip_addr* [*mask*] | {**object** | **object-group**} *name*] [**translated** [**interface** *name*] {*ip_addr* [*mask*] | {**object** | **object-group**} *name*}] [**detail**]

Syntax Description

detail	(Optional) Includes more verbose expansion of the object fields.
interface *name*	(Optional) Specifies the source interface.
ip_addr [*mask*]	(Optional) Specifies an IP address and subnet mask.
object *name*	(Optional) Specifies a network object or service object.
object-group *name*	(Optional) Specifies a network object group
translated	(Optional) Specifies the translated parameters.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **show nat** command to show runtime representation of the NAT policy. Use the **detail** optional keyword to expand the object and view the object values. Use the additional selector fields to limit the **show nat** command output.

The output shows all NAT commands, even hidden commands. For example, if you configure the management interface to use the data interfaces as a gateway, hidden NAT rules are created for a hidden virtual interface (for example, nlp_int_tap) to enable communications between the management interface and each data interface. These rules are not reflected in the NAT tables in device manager. You will also see hidden rules for any HTTPS/SSH management access rules that allow management connections to data interfaces, which are reflected in the device manager's management access table but not in the NAT table. Starting in version 7.0, any rules the system creates for its own use are listed in Section 0.

Examples

The following is sample output from the **show nat** command:

```
> show nat
   Manual NAT Policies (Section 1)
   1 (any) to (any) source dynamic S S' destination static D' D
       translate_hits = 0, untranslate_hits = 0

   Auto NAT Policies (Section 2)
   1 (inside) to (outside) source dynamic A 2.2.2.2
       translate_hits = 0, untranslate_hits = 0

   Manual NAT Policies (Section 3)
   1 (any) to (any) source dynamic C C' destination static B' B service R R'
       translate_hits = 0, untranslate_hits = 0
```

```
> show nat detail
   Manual NAT Policies (Section 1)
   1 (any) to (any) source dynamic S S' destination static D' D
       translate_hits = 0, untranslate_hits = 0
       Source - Real: 1.1.1.2/32, Mapped: 2.2.2.3/32
       Destination - Real: 10.10.10.0/24, Mapped: 20.20.20.0/24

   Auto NAT Policies (Section 2)
   1 (inside) to (outside) source dynamic A 2.2.2.2
       translate_hits = 0, untranslate_hits = 0
       Source - Real: 1.1.1.1/32, Mapped: 2.2.2.2/32

   Manual NAT Policies (Section 3)
   1 (any) to (any) source dynamic C C' destination static B' B service R R'
       translate_hits = 0, untranslate_hits = 0
       Source - Real: 11.11.11.10-11.11.11.11, Mapped: 192.168.10.10/32
       Destination - Real: 192.168.1.0/24, Mapped: 10.75.1.0/24
       Service - Real: tcp source eq 10 destination eq ftp-data , Mapped: tcp source eq
       100 destination eq 200
```

The following is sample output from the **show nat detail** command between IPv6 and IPv4:

```
> show nat detail
1 (in) to (outside) source dynamic inside_nw outside_map destination static inside_map any
translate_hits = 0, untranslate_hits = 0
Source - Origin: 2001::/96, Translated: 192.168.102.200-192.168.102.210
Destination - Origin: 2001::/96, Translated: 0.0.0.0/0
```

The following example shows system-defined rules in section 0.

```
> show nat detail
Manual NAT Policies Implicit (Section 0)
1 (nlp_int_tap) to (inside) source static nlp_server_0_snmp_intf3 interface  service udp
snmp snmp
    translate_hits = 1, untranslate_hits = 1
    Source - Origin: 169.254.1.2/32, Translated: 10.1.1.122/24
    Service - Protocol: udp Real: snmp Mapped: snmp
2 (nlp_int_tap) to (inside) source dynamic nlp_client_0_intf3 interface
    translate_hits = 0, untranslate_hits = 0
    Source - Origin: 169.254.1.2/32, Translated: 10.1.1.122/24

Manual NAT Policies (Section 1)
1 (inside) to (any) source dynamic obj_man interface
    translate_hits = 0, untranslate_hits = 0
    Source - Origin: 10.3.3.3/32, Translated: 10.1.1.122/24
```

Related Commands

Command	Description
clear nat counters	Clears NAT policy counters.

show nat divert-table

To display statistics of NAT divert table, use the **show nat divert-table** command.

show nat divert-table [**ipv6**] [**interface** *interface_name*]

Syntax Description

divert-table	Shows the NAT divert table.
ipv6	(Optional) Shows IPv6 entries in the divert table.
interface *interface_name*	(Optional) Limits output to the specified source interface.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **show nat divert-table** command to show runtime representation of the NAT divert table. Use the **ipv6** optional keyword to view the IPv6 entries in the divert table. Use the **interface** optional keyword to view the NAT divert table for the specific source interface.

The divert table shows all NAT commands, even hidden commands. For example, if you configure the management interface to use the data interfaces as a gateway, hidden NAT rules are created for a hidden virtual interface (for example, nlp_int_tap) to enable communications between the management interface and each data interface. These rules are not reflected in the NAT tables in device manager.

Examples

The following is sample output from the **show nat divert-table** command:

```
> show nat divert-table
Divert Table
id=0xad1521b8, domain=twice-nat section=1 ignore=no
        type=none, hits=0, flags=0x9, protocol=0
        src ip/id=0.0.0.0, mask=0.0.0.0, port=0-0
        dst ip/id=10.86.119.255, mask=255.255.255.255, port=0-0
        input_ifc=outside, output_ifc=NP Identity Ifc
id=0xad1523a8, domain=twice-nat section=1 ignore=no
        type=none, hits=0, flags=0x9, protocol=0
        src ip/id=0.0.0.0, mask=0.0.0.0, port=0-0
        dst ip/id=10.86.116.0, mask=255.255.255.255, port=0-0
        input_ifc=outside, output_ifc=NP Identity Ifc
id=0xad1865c0, domain=twice-nat section=1 ignore=no
        type=none, hits=0, flags=0x9, protocol=0
        src ip/id=0.0.0.0, mask=0.0.0.0, port=0-0
        dst ip/id=192.168.255.255, mask=255.255.255.255, port=0-0
        input_ifc=amallio-wizard, output_ifc=NP Identity Ifc
id=0xad1867b0, domain=twice-nat section=1 ignore=no
        type=none, hits=0, flags=0x9, protocol=0
        src ip/id=0.0.0.0, mask=0.0.0.0, port=0-0
        dst ip/id=192.168.0.0, mask=255.255.255.255, port=0-0
        input_ifc=amallio-wizard, output_ifc=NP Identity Ifc
id=0xad257bf8, domain=twice-nat section=1 ignore=no
        type=none, hits=0, flags=0x9, protocol=0
```

```
        src ip/id=0.0.0.0, mask=0.0.0.0, port=0-0
        dst ip/id=172.27.48.255, mask=255.255.255.255, port=0-0
        input_ifc=folink, output_ifc=NP Identity Ifc
id=0xad257db8, domain=twice-nat section=1 ignore=no
        type=none, hits=0, flags=0x9, protocol=0
        src ip/id=0.0.0.0, mask=0.0.0.0, port=0-0
        dst ip/id=172.27.48.0, mask=255.255.255.255, port=0-0
        input_ifc=folink, output_ifc=NP Identity Ifc
```

The following is sample output from the **show nat divert ipv6** command:

```
> show nat divert ipv6
Divert Table
id=0xcb9ea518, domain=divert-route
type=static, hits=0, flags=0x21, protocol=0
src ip/id=2001::/ffff:ffff:ffff:ffff:ffff:ffff::, port=0-0
dst ip/id=2001::/ffff:ffff:ffff:ffff:ffff:ffff::, port=0-0
input_ifc=in, output_ifc=outside
id=0xcf24d4b8, domain=divert-route
type=static, hits=0, flags=0x20, protocol=0
src ip/id=::/::, port=0-0
dst ip/id=2222::/ffff:ffff:ffff:ffff:ffff:ffff::, port=0-0
input_ifc=in, output_ifc=mgmt
```

Related Commands

Command	Description
clear nat counters	Clears NAT policy counters.
show nat	Displays runtime representation of the NAT policies.

show nat pool

To display statistics of NAT pool usage, use the **show nat pool** command.

show nat pool [**interface** *if-name* [**ip** *address*] | **ip** *address* | **detail**]

show nat pool cluster [**summary** | **interface** *if-name* [**ip** *address*] | **ip** *address*]

Syntax Description

cluster	(Optional) When clustering is enabled, shows the current assignment of a PAT address to the owner unit and backup unit. (6.7+) Include the **summary** keyword to see the distribution of port blocks among the units in the cluster.
interface *if_name*	Limit the display to pools for the named interface. You can optionally include the **ip** keyword to further limit the view.
ip *address*	Limit the display to the specified IP address from the PAT pool.
detail	Show information related to the usage and distribution of port blocks within a cluster. This keyword appears only if the unit is a cluster member. You cannot use it with the cluster keyword.

Command History

Release	Modification
6.1	This command was introduced.
6.7	The following keywords were added: **interface**, **ip**, **detail**, **summary**.

Usage Guidelines

(Pre-6.7) A NAT pool is created for each mapped protocol/IP address/port range, where the port ranges are 1-511, 512-1023, and 1024-65535 by default. If you configure the PAT pool to use a flat range of ports, you will see fewer, larger ranges.

(6.7+) Starting with 6.7, the port range is flat by default, and you can optionally include the reserved ports, 1-1023, in the pool. For clustered systems, the PAT pool is distributed among the cluster members in blocks of 512 ports.

Each NAT pool exists for at least 10 minutes after the last usage. The 10 minute hold-down timer is canceled if you clear the translations with **clear xlate**.

Examples

The following is sample output for the NAT pools created by a dynamic PAT rule shown by the **show running-config object network** command.

```
> show running-config object network
object network myhost
 host 10.10.10.10
 nat (pppoe2,inside) dynamic 10.76.11.25

> show nat pool
```

```
TCP inside, address 10.76.11.25, range 1-511, allocated 0
TCP inside, address 10.76.11.25, range 512-1023, allocated 0
TCP inside, address 10.76.11.25, range 1024-65535, allocated 1
```

The following is sample output from the **show nat pool** command showing use of the PAT pool **flat** option. Without the **include-reserve** keyword, two ranges are shown; the lower range is used when a source port below 1024 is mapped to the same port.

```
> show nat pool
ICMP PAT pool dynamic-pat, address 172.16.2.200, range 1-65535, allocated 2
TCP PAT pool dynamic-pat, address 172.16.2.200, range 1-1024, allocated 0
TCP PAT pool dynamic-pat, address 172.16.2.200, range 1024-65535, allocated 2
UDP PAT pool dynamic-pat, address 172.16.2.200, range 1-1024, allocated 0
UDP PAT pool dynamic-pat, address 172.16.2.200, range 1024-65535, allocated 2
```

The following is sample output from the **show nat pool** command showing use of the PAT pool **flat include-reserve** options.

```
> show nat pool
ICMP PAT pool dynamic-pat, address 172.16.2.200, range 1-65535, allocated 2
TCP PAT pool dynamic-pat, address 172.16.2.200, range 1-65535, allocated 2
UDP PAT pool dynamic-pat, address 172.16.2.200, range 1-65535, allocated 2
```

(Pre-6.7) The following is sample output from the **show nat pool** command showing use of the PAT pool **extended flat include-reserve** options. The important items are the parenthetical addresses. These are the destination addresses used to extend PAT.

```
ICMP PAT pool dynamic-pat, address 172.16.2.200, range 1-65535, allocated 0
ICMP PAT pool dynamic-pat, address 172.16.2.200(172.16.2.99), range 1-65535,
allocated 2
TCP PAT pool dynamic-pat, address 172.16.2.200(172.16.2.100), range 1-65535,
allocated 1
UDP PAT pool dynamic-pat, address 172.16.2.200(172.16.2.100), range 1-65535,
allocated 1
TCP PAT pool dynamic-pat, address 172.16.2.200, range 1-65535, allocated 0
ICMP PAT pool dynamic-pat, address 172.16.2.200(172.16.2.100), range 1-65535,
allocated 1
TCP PAT pool dynamic-pat, address 172.16.2.200(172.16.2.99), range 1-65535,
allocated 2
UDP PAT pool dynamic-pat, address 172.16.2.200, range 1-65535, allocated 0
```

(6.7+) The following example shows the distribution of port blocks (showing the port range), and their usage, in a cluster, including the unit that owns the block and the backup unit for the block.

```
> show nat pool cluster
IP outside_a:src_map_a 174.0.1.20
               [1536 - 2047], owner A, backup B
               [8192 - 8703], owner A, backup B
               [4089 - 4600], owner B, backup A
               [11243 - 11754], owner B, backup A
IP outside_a:src_map_a 174.0.1.21
               [1536 - 2047], owner A, backup B
               [8192 - 8703], owner A, backup B
               [4089 - 4600], owner B, backup A
               [11243 - 11754], owner B, backup A
IP outside_b:src_map_b 174.0.1.22
```

```
                [6656 - 7167], owner A, backup B
                [13312 - 13823], owner A, backup B
                [20480 - 20991], owner B, backup A
                [58368 - 58879], owner B, backup A
IP outside_b:src_map_b 174.0.1.23
                [46592 - 47103], owner A, backup B
                [52224 - 52735], owner A, backup B
                [62976 - 63487], owner B, backup A
```

(6.7+) The following example shows a summary of pool assignments in a cluster.

```
> show nat pool cluster summary
port-blocks count display order: total, unit-A, unit-B, unit-C, unit-D
IP outside_a:src_map_a, 174.0.1.20 (128 - 32/32/32/32)
IP outside_a:src_map_a, 174.0.1.21 (128 - 36/32/32/28)
IP outside_b:src_map_b, 174.0.1.22 (128 - 31/32/32/33)
```

(7.0+) The following example shows a summary of pool assignments in a cluster. Starting with 7.0, the information includes the number of reserved ports and reclaimed ports.

```
> show nat pool cluster summary

port-blocks count display order: total, unit-A, unit-B
Codes: ^ - reserve, # - reclaimable
IP Outside:Mapped-IPGroup 10.10.10.100 (126 - 63 / 63) ^ 0 # 0
IP Outside:Mapped-IPGroup 10.10.10.101 (126 - 63 / 63) ^ 0 # 0
```

(6.7+) The following example shows detailed PAT pool usage for the pools in a cluster. When viewing detailed output, backup port ranges are indicated with an asterisk. For example: range 62464-62975, allocated 27 *

```
> show nat pool detail
TCP PAT pool outside_a, address 174.0.1.1
                range 1536-2047, allocated 56
                range 8192-8703, allocated 16
UDP PAT pool outside_a, address 174.0.1.1
                range 1536-2047, allocated 12
                range 8192-8703, allocated 25
 TCP PAT pool outside_b, address 174.0.2.1
                range 47104-47615, allocated 39
                range 62464-62975, allocated 9
UDP PAT pool outside_b, address 174.0.2.1
                range 47104-47615, allocated 35
                range 62464-62975, allocated 27*
```

(6.7+) The following example shows how to limit the view to a specific interface on a specific device.

```
> show nat pool interface outside_b ip 174.0.2.1
TCP PAT pool outside_b, address 174.0.2.1, range 1-511, allocated 0
TCP PAT pool outside_b, address 174.0.2.1, range 512-1023, allocated 12
TCP PAT pool outside_b, address 174.0.2.1, range 1024-65535, allocated 48
UDP PAT pool outside_b, address 174.0.2.1, range 1-511, allocated 6
UDP PAT pool outside_b, address 174.0.2.1, range 512-1023, allocated 8
UDP PAT pool outside_b, address 174.0.2.1, range 1024-65535, allocated 62
```

Related Commands

Command	Description
show nat	Displays NAT policy statistics.

show nat proxy-arp

To display the NAT proxy ARP table, use the **show nat proxy-arp** command.

show nat proxy-arp [**ipv6**] [**interface** *name*]

Syntax Description

ipv6	(Optional) Shows IPv6 entries in the proxy ARP table.
interface *name*	(Optional) Limits output to the specified source interface.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **show nat proxy-arp** command to show runtime representation of the NAT proxy ARP table.

The proxy ARP table shows all NAT commands, even hidden commands. For example, if you configure the management interface to use the data interfaces as a gateway, hidden NAT rules are created for a hidden virtual interface (for example, nlp_int_tap) to enable communications between the management interface and each data interface. These rules are not reflected in the NAT tables in device manager.

Examples

The following is sample output from the **show nat proxy-arp** command:

```
> show nat proxy-arp
Nat Proxy-arp Table
id=0x00007f4ce491a010, ip/id=0.0.0.0, mask=255.255.255.255 ifc=outside
        config:(inside1_8) to (outside) source dynamic any-ipv4 interface
id=0x00007f4cdc6138d0, ip/id=0.0.0.0, mask=255.255.255.255 ifc=outside
        config:(inside1_7) to (outside) source dynamic any-ipv4 interface
id=0x00007f4ce491d2e0, ip/id=0.0.0.0, mask=255.255.255.255 ifc=outside
        config:(inside1_6) to (outside) source dynamic any-ipv4 interface
id=0x00007f4cdc618a10, ip/id=0.0.0.0, mask=255.255.255.255 ifc=outside
        config:(inside1_5) to (outside) source dynamic any-ipv4 interface
id=0x00007f4d019c9e70, ip/id=0.0.0.0, mask=255.255.255.255 ifc=outside
        config:(inside1_4) to (outside) source dynamic any-ipv4 interface
id=0x00007f4cdc61b300, ip/id=0.0.0.0, mask=255.255.255.255 ifc=outside
        config:(inside1_3) to (outside) source dynamic any-ipv4 interface
id=0x00007f4ce49261f0, ip/id=0.0.0.0, mask=255.255.255.255 ifc=outside
        config:(inside1_2) to (outside) source dynamic any-ipv4 interface
```

Related Commands

Command	Description
clear nat counters	Clears NAT policy counters.
show nat	Displays runtime representation of the NAT policies.

show network

To display the attributes of the management interface, use the **show network** command.

show network

Command History

Release	Modification
6.1	This command was introduced.
6.7	This command now shows both Management and management center access data interface network settings.

Usage Guidelines

Use this command to view the management interface properties, which you set using the **configure network** commands.

If you configure the management address to use the data interfaces as the gateway, the Gateway is shown as "data-interface."

Examples

The following is sample output for the **show network** command.

```
> show network
===============[ System Information ]===============
Hostname                : 5516X-4
DNS Servers             : 208.67.220.220,208.67.222.222
Management port         : 8305
IPv4 Default route
  Gateway               : data-interfaces
IPv6 Default route
  Gateway               : data-interfaces

======================[ br1 ]=======================
State                   : Enabled
Link                    : Up
Channels                : Management & Events
Mode                    : Non-Autonegotiation
MDI/MDIX                : Auto/MDIX
MTU                     : 1500
MAC Address             : 28:6F:7F:D3:CB:8D
----------------------[ IPv4 ]----------------------
Configuration           : Manual
Address                 : 10.99.10.4
Netmask                 : 255.255.255.0
Gateway                 : 10.99.10.1
----------------------[ IPv6 ]----------------------
Configuration           : Disabled

===============[ Proxy Information ]================
State                   : Disabled
Authentication          : Disabled

======[ System Information - Data Interfaces ]======
DNS Servers             :
Interfaces              : GigabitEthernet1/1
```

```
===============[ GigabitEthernet1/1 ]===============
State                     : Enabled
Link                      : Up
Name                      : outside
MTU                       : 1500
MAC Address               : 28:6F:7F:D3:CB:8F
----------------------[ IPv4 ]----------------------
Configuration             : Manual
Address                   : 10.89.5.29
Netmask                   : 255.255.255.192
Gateway                   : 10.89.5.1
----------------------[ IPv6 ]----------------------
Configuration             : Disabled
```

show network-dhcp-server

To display the status of the DHCP server on the management interface, use the **show network-dhcp-server** command.

show network-dhcp-server

Command History

Release	Modification
6.2	This command was introduced.

Usage Guidelines

Use this command to view the status of the optional DHCP server for the management interface. To configure the DHCP server, use the **configure network ipv4 dhcp-server-enable** command.

The output shows whether the DHCP server is enabled or disabled. If enabled, it also shows the address pool.

Examples

The following example shows how to configure the DHCP server and show its status.

```
> show network-dhcp-server
DHCP Server Disabled
> configure network ipv4 dhcp-server-enable 192.168.45.46 192.168.45.254
DHCP Server Enabled
> show network-dhcp-server
DHCP Server Enabled
192.168.45.46-192.168.45.254
```

Related Commands

Command	Description
configure network ipv4 dhcp-server-enable	Configures the DHCP server on the management interface.
configure network ipv4 dhcp-server-disable	Disables the DHCP server on the management interface.

show network-static-routes

To display static routes configured for the management interface, use the **show network-static-routes** command.

show network-static-routes

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Static routes for the management interface are used when you configure multiple management interfaces. These routes do not include the default gateway. If you are using a single management interface, you typically would not have additional static routes.

The routes shown with this command are for the management interface only. They are not used by any data interface. They are not used for through-the-box traffic.

Examples

The following example shows that there are no additional static routes for the management interface. The default gateway is the only route.

```
> show network-static-routes
No static routes currently configured.
```

The following example shows one static route.

```
> show network-static-routes
---------------[ IPv4 Static Routes ]---------------
Interface                : br1
Destination              : 10.1.1.0
Gateway                  : 192.168.0.254
Netmask                  : 255.255.255.0
```

Related Commands

Command	Description
configure network static-routes	Configure static routes for the management interface.

show ntp

To display the current Network Time Protocol (NTP) servers and configuration, use the **show ntp** command.

show ntp

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

This command displays basic information about the NTP servers. If you need more extensive information, use the **system support ntp** command, which includes the output from this command plus the output from the standard NTP command **ntpq**, which is documented with the NTP protocol.

Examples

The following example shows how to display the NTP configuration.

```
> show ntp
NTP Server                : 209.208.79.69
Status                    : Available
Offset                    : -1.614 (milliseconds)
Last Update               : 578 (seconds)

NTP Server                : 45.127.112.2  (clocka.ntpjs.org)
Status                    : Available
Offset                    : -1.355 (milliseconds)
Last Update               : 874 (seconds)

NTP Server                : 198.58.105.63  (ha81.smatwebdesign.com)
Status                    : Not Available
Offset                    : -4.942 (milliseconds)
Last Update               : 369 (seconds)

NTP Server                : 204.9.54.119  (ntp.your.org)
Status                    : Being Used
Offset                    : 0.312 (milliseconds)
Last Update               : 962 (seconds)
```

The following example shows how to use the **system support ntp** command to get additional information. Use this command if you need to confirm NTP synchronization.

Look for the section “Results of ‘ntpq -pn.’ For example, you might see something like the following:

```
> system support ntp
... output redacted ...
Results of 'ntpq -pn'
remote                    : +216.229.0.50
refid                     : 129.7.1.66
st                        : 2
t                         : u
when                      : 704
poll                      : 1024
reach                     : 377
delay                     : 90.455
```

```
offset                    : 2.954
jitter                    : 2.473
... remaining output redacted ...
```

In this example, the + before the NTP server address indicates that it is a potential candidate. An asterisk here, *, indicates the current time source peer.

The NTP daemon (NTPD) uses a sliding window of eight samples from each one of the peers and picks out one sample, then the clock selection determines the true chimers and the false tickers. NTPD then determines the round-trip distance (the offset of a candidate must not be over one-half the round trip delay). If connection delays, packet loss, or server issues cause one or all the candidates to be rejected, you would see long delays in the synchronization. The adjustment also occurs over a very long period of time: the clock offset and oscillator errors must be resolved by the clock discipline algorithm and this can take hours.

Note If the refid is .LOCL., this indicates the peer is an undisciplined local clock, that is, it is using its local clock only to set the time. device manager always marks the NTP connection yellow (not synchronized) if the selected peer is .LOCL. Normally, NTP does not select a .LOCL. candidate if a better one is available, which is why you should configure at least three servers.

Related Commands

Command	Description
system support ntp	Shows detailed troubleshooting information for NTP.

show object

To display information about network-service objects, including hit counts and IP addresses, use the **show object** command.

show object [**id** *object_name* | **network-service** [**detail**]]

Syntax Description

id *name*	(Optional) The name of the object you want to view. Capitalization matters. For example “object-name” does not match “Object-Name.”
network-service [**detail**]	(Optional.) Show all network-service objects. Include the detail keyword to see the cached IP addresses associated with the object members.

Command Default

Without parameters, all objects are shown.

Command History

Release	Modification
7.1	This command was introduced.

Example

The following example shows the details for the network-service object named Cisco. The app-id (application ID) is an internal number. The hitcnt (hit count) number is the only relevant metric shown.

```
> show object id Cisco
object network-service "Cisco" dynamic
 description Official website for Cisco.
 app-id 2655
 domain cisco.com (bid=0) ip (hitcnt=0)
```

Related Commands

Command	Description
clear object	Clears the network-service objects hit count.
show object-groups	Shows network-service object groups and hit counts.

show object-group

To display object group information and the relevant hit count if the object group is of the network or network-service object-group type, use the **show object-group** command. Use the command without parameters to see all types of object group.

show object-group [**count** | **interface** | **network** | **security** | **service** | **id** *name*]

show object-group network-service [*group_name* [**network-service-member** *member_name* [**dns** *domain_name*]] [**detail**]

Syntax Description

count	(Optional.) Show statistics related to the number of object groups and the number of objects in those groups, and how they are used.
detail	For network-service objects, show the cached IP addresses associated with the object members.
dns *domain_name*	(Optional.) For network-service objects specified by name and member, limit the information to a specific domain for that member. For example, example.com.
id *name*	(Optional) Identifies an object group by name.
interface	(Optional) Interface-type objects
network	(Optional) Network-type objects.
network-service [*group_name*]	(Optional.) Network-service objects. You can specify the object name to limit the information to a single object.
network-service-member *member_name*	(Optional.) For network-service objects specified by name, limit the information to a specific member of that object.
security	(Optional) Security-type objects
service	(Optional) Service-type objects.

Command History

Release	Modification
6.1	This command was introduced.
7.1	We added the **network-service** keyword and its associated parameters.
7.2	The **count** keyword was added.

Examples

The following is sample output from the **show object-group** command and shows information about the network object group named “Anet”:

```
> show object-group id Anet
```

```
Object-group network Anet (hitcnt=10)
   Description OBJ SEARCH ALG APPLIED
   network-object 1.1.1.0 255.255.255.0 (hitcnt=4)
   network-object 2.2.2.0 255.255.255.0 (hitcnt=6)
```

The following is sample output from the **show object-group** command and shows information about a service group:

```
> show object-group service
object-group service B-Serobj
   description its a service group
   service-object tcp eq bgp
```

The following example shows a network-service object and its hit counts. The various identifiers, such as network-service group ID (nsg-id), application ID (app-id), and bid are internal indexing numbers that you can ignore.

```
> show object-group network-service FMC_NSG_4294969442
object-group network-service FMC_NSG_4294969442 (nsg-id 512/1)
 network-service-member "Facebook" dynamic
  description Facebook is a social networking service.
  app-id 629
  domain connect.facebook.net (bid=214491) ip (hitcnt=0)
  domain facebook.com (bid=370809) ip (hitcnt=0)
  domain fbcdn.net (bid=490321) ip (hitcnt=0)
  domain fbcdn-photos-a.akamaihd.net (bid=548791) ip (hitcnt=0)
  domain fbcdn-photos-e-a.akamaihd.net (bid=681143) ip (hitcnt=0)
  domain fbcdn-photos-b-a.akamaihd.net (bid=840741) ip (hitcnt=0)
  domain fbstatic-a.akamaihd.net (bid=1014669) ip (hitcnt=0)
  domain fbexternal-a.akamaihd.net (bid=1098051) ip (hitcnt=0)
  domain fbcdn-profile-a.akamaihd.net (bid=1217875) ip (hitcnt=0)
  domain fbcdn-creative-a.akamaihd.net (bid=1379985) ip (hitcnt=0)
  domain channel.facebook.com (bid=1524617) ip (hitcnt=0)
  domain fbcdn-dragon-a.akamaihd.net (bid=1683343) ip (hitcnt=0)
  domain contentcache-a.akamaihd.net (bid=1782703) ip (hitcnt=0)
  domain facebook.net (bid=1868733) ip (hitcnt=0)
 network-service-member "Google+ Videos" dynamic
  description Video sharing among Google+ community.
  app-id 2881
  domain plus.google.com (bid=2068293) ip (hitcnt=0)
 network-service-member "Instagram" dynamic
  description Mobile phone photo sharing.
  app-id 1233
  domain instagram.com (bid=2176667) ip (hitcnt=0)
 network-service-member "LinkedIn" dynamic
  description Career oriented social networking.
  app-id 713
  domain linkedin.com (bid=2317259) ip (hitcnt=0)
>
```

The following example shows object counts, so you have an idea of how many object groups there are, how many objects are contained in the groups, and how many are used in ACLs, NAT, and so forth. This information relates to the performance of the object group search feature.

```
ciscoasa(config)# show object-group count

Object Group Name                     Group Count Dyn Count V4 CNT     V6 CNT     ACL CNT
    NAT CNT    OG in OG
network    i28Z-route                 68          0           68         0          0
      0          0
```

```
network    i28Z-VRF-BGP-PEERS          4           0           4           0           2
      0          0
network    EXCH-BGP-PEERS              4           0           4           0           2
      0          0
network    obgr_SUBNETS_NO_ACL         112         0           112         0           0
      0          0
network    obgr_SUBNETS_ACL_ASAMgmt    1           0           1           0           0
      0          0
network    obgr_CLIENTS_ACL_ASAMgmt    8           0           8           0           1
      0          0
network    obgr_SUBNETS_CGS_vMotion    1           0           1           0           0
      0          0
network    obgr_CLIENTS_CGS_vMotion    9           0           9           0           1
      0          0
network    obgr_SUBNETS_UPMCOD_CGS     17          0           17          0           0
      0          0
network    obgr_CLIENTS_UPMCOD_CGS     90          0           90          0           1
      0          0
network    obgr_CLIENTS_10.68.0.0_16   2           0           2           0           1
      0          0
network    obgr_CLIENTS_10.68.1.198_31 4           0           4           0           1
      0          0
network    obgr_CLIENTS_10.68.73.133   7           0           7           0           1
      0          0
network    asa_zabbix_proxies          4           0           4           0           1
      0          0

Total Summary
Object-group count                          14
Object-group object count                   331
Object-group Dynamic count                  0
Object-group IPv4 count                     331
Object-group IPv6 count                     0
Object-group Used in ACL                    9
Object-group Used in NAT                    0
Object-group Unused                         5
Object-group Internal                       0
Object-group Dummy                          0
Redundant object-group in Network           4
Redundant object-group in IfC               0
```

Related Commands

Command	Description
clear object-group	Clears the network objects hit count for a given object group.
show access-list	Shows all access lists, relevant expanded access list entries, and hit counts.
show object	Shows network-service objects and hit counts.

show ospf

To display the general information about the OSPF routing processes, use the **show ospf** command.

show ospf [**vrf** *name* | **all**] [*pid* [*area_id*]]

Syntax Description

area_id	(Optional) ID of the area that is associated with the OSPF address range.
pid	(Optional) The ID of the OSPF process.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf** command, showing how to display general information about a specific OSPF routing process:

```
> show ospf 5
 Routing Process "ospf 5" with ID 127.0.0.1 and Domain ID 0.0.0.5
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x     0
 Number of opaque AS LSA 0. Checksum Sum 0x     0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0
```

The following is sample output from the **show ospf** command, showing how to display general information about all OSPF routing processes:

```
> show ospf
 Routing Process "ospf 5" with ID 127.0.0.1 and Domain ID 0.0.0.5
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x     0
 Number of opaque AS LSA 0. Checksum Sum 0x     0
 Number of DCbitless external and opaque AS LSA 0
```

```
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 0. 0 normal 0 stub 0 nssa
External flood list length 0

Routing Process "ospf 12" with ID 172.23.59.232 and Domain ID 0.0.0.12
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x     0
Number of opaque AS LSA 0. Checksum Sum 0x     0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 0. 0 normal 0 stub 0 nssa
External flood list length 0
```

show ospf border-routers

To display the internal OSPF routing table entries to ABRs and ASBRs, use the **show ospf border-routers** command.

show ospf border-routers [**vrf** *name* | **all**]

Syntax Description

[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf border-routers** command:

```
> show ospf border-routers

OSPF Process 109 internal Routing Table

Codes: i - Intra-area route, I - Inter-area route

i 192.168.97.53 [10] via 192.168.1.53, fifth, ABR, Area 0, SPF 20
i 192.168.103.51 [10] via 192.168.96.51, outside, ASBR, Area 192.168.12.0, SPF 14
i 192.168.103.52 [10] via 192.168.96.51, outside, ABR/ASBR, Area 192.168.12.0, SPF 14
```

show ospf database

To display the information contained in the OSPF topological database, use the **show ospf database** command.

show ospf [**vrf** *name* | **all**] [*pid* [*area_id*]] **database** [**router** | **network** | **summary** | **asbr-summary** | **external** | **nssa-external**] [*lsid*] [**internal**] [**self-originate** | **adv-router** *addr*]
show ospf [*pid* [*area_id*]] **database database-summary**

Syntax Description

addr	(Optional) Router address.
adv-router	(Optional) Advertised router.
area_id	(Optional) ID of the area that is associated with the OSPF address range.
asbr-summary	(Optional) Displays an ASBR list summary.
database	Displays the database information.
database-summary	(Optional) Displays the complete database summary list.
external	(Optional) Displays routes external to a specified autonomous system.
internal	(Optional) Routes that are internal to a specified autonomous system.
lsid	(Optional) LSA ID.
network	(Optional) Displays the OSPF database information about the network.
nssa-external	(Optional) Displays the external not-so-stubby-area list.
pid	(Optional) ID of the OSPF process.
router	(Optional) Displays the router.
self-originate	(Optional) Displays the information for the specified autonomous system.
summary	(Optional) Displays a summary of the list.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf database** command:

```
> show ospf database
OSPF Router with ID(192.168.1.11) (Process ID 1)

                Router Link States(Area 0)
Link ID   ADV Router   Age   Seq# Checksum Link count
192.168.1.8 192.168.1.8 1381 0x8000010D    0xEF60 2
192.168.1.11 192.168.1.11 1460 0x800002FE    0xEB3D 4
192.168.1.12 192.168.1.12 2027 0x80000090    0x875D 3
192.168.1.27 192.168.1.27 1323 0x800001D6    0x12CC 3

                Net Link States(Area 0)
Link ID ADV Router   Age   Seq# Checksum
172.16.1.27 192.168.1.27 1323 0x8000005B    0xA8EE
172.17.1.11 192.168.1.11 1461 0x8000005B    0x7AC

                Type-10 Opaque Link Area Link States (Area 0)
Link ID ADV Router   Age Seq# Checksum Opaque ID
10.0.0.0 192.168.1.11 1461 0x800002C8    0x8483   0
10.0.0.0 192.168.1.12 2027 0x80000080    0xF858   0
10.0.0.0 192.168.1.27 1323 0x800001BC    0x919B   0
10.0.0.1 192.168.1.11 1461 0x8000005E    0x5B43   1
```

The following is sample output from the **show ospf database asbr-summary** command:

```
> show ospf database asbr-summary
OSPF Router with ID(192.168.239.66) (Process ID 300)
Summary ASB Link States(Area 0.0.0.0)
Routing Bit Set on this LSA
LS age: 1463
Options: (No TOS-capability)
LS Type: Summary Links(AS Boundary Router)
Link State ID: 172.16.245.1 (AS Boundary Router address)
Advertising Router: 172.16.241.5
LS Seq Number: 80000072
Checksum: 0x3548
Length: 28
Network Mask: 0.0.0.0
TOS: 0 Metric: 1
```

The following is sample output from the **show ospf database router** command:

```
> show ospf database router
OSPF Router with id(192.168.239.66) (Process ID 300)
Router Link States(Area 0.0.0.0)
Routing Bit Set on this LSA
LS age: 1176
Options: (No TOS-capability)
LS Type: Router Links
Link State ID: 10.187.21.6
Advertising Router: 10.187.21.6
LS Seq Number: 80002CF6
Checksum: 0x73B7
Length: 120
AS Boundary Router
Number of Links: 8
```

```
Link connected to: another Router (point-to-point)
(link ID) Neighboring Router ID: 10.187.21.5
(Link Data) Router Interface address: 10.187.21.6
Number of TOS metrics: 0
TOS 0 Metrics: 2
```

The following is sample output from the **show ospf database network** command:

```
> show ospf database network
OSPF Router with id(192.168.239.66) (Process ID 300)
Displaying Net Link States(Area 0.0.0.0)
LS age: 1367
Options: (No TOS-capability)
LS Type: Network Links
Link State ID: 10.187.1.3 (address of Designated Router)
Advertising Router: 192.168.239.66
LS Seq Number: 800000E7
Checksum: 0x1229
Length: 52
Network Mask: 255.255.255.0
Attached Router: 192.168.239.66
Attached Router: 10.187.241.5
Attached Router: 10.187.1.1
Attached Router: 10.187.54.5
Attached Router: 10.187.1.5
```

The following is sample output from the **show ospf database summary** command:

```
> show ospf database summary
OSPF Router with id(192.168.239.66) (Process ID 300)
Displaying Summary Net Link States(Area 0.0.0.0)
LS age: 1401
Options: (No TOS-capability)
LS Type: Summary Links(Network)
Link State ID: 10.187.240.0 (summary Network Number)
Advertising Router: 10.187.241.5
LS Seq Number: 80000072
Checksum: 0x84FF
Length: 28
Network Mask: 255.255.255.0 TOS: 0 Metric: 1
```

The following is sample output from the **show ospf database external** command:

```
> show ospf database external
OSPF Router with id(192.168.239.66) (Autonomous system 300)

                  Displaying AS External Link States
LS age: 280
Options: (No TOS-capability)
LS Type: AS External Link
Link State ID: 172.16.0.0 (External Network Number)
Advertising Router: 10.187.70.6
LS Seq Number: 80000AFD
Checksum: 0xC3A
Length: 36
Network Mask: 255.255.0.0

      Metric Type: 2 (Larger than any link state path)
TOS: 0
Metric: 1
```

```
Forward Address: 0.0.0.0
External Route Tag: 0
```

show ospf events

To display OSPF internal event information, use the **show ospf events** command.

show ospf [**vrf** *name* | **all**] [*process_id*] **events** [*type*]

Syntax Description

process_id	(Optional) Specifies an internal ID that is locally assigned and can be any positive integer. This ID is the number assigned administratively when the OSPF routing process is enabled.
type	(Optional) A list of the event types you want to see. If you do not specify one or more types, you see all events. You can filter on the following types: • **generic**—Generic events. • **interface**—Interface state change events. • **lsa**—LSA arrival and LSA generation events. • **neighbor**—Neighbor state change events. • **reverse**—Show events in reverse order. • **rib**—Router Information Base update, delete and redistribution events. • **spf**—SPF scheduling and SPF run events.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf events** command:

```
> show ospf events

           OSPF Router with ID (192.168.77.1) (Process ID 5)

  1 Apr 27 16:33:23.556: RIB Redist, dest 0.0.0.0, mask 0.0.0.0, Up
  2 Apr 27 16:33:23.556: Rescanning RIB:  0x00x0
  3 Apr 27 16:33:23.556: Service Redist scan:  0x00x0
```

Related Commands

Command	Description
show ospf	Shows all settings in the OSPF routing process.
show ospf border-routers	Shows the internal OSPF routing table entries to an area border router (ABR) and an autonomous system boundary router (ASBR).

show ospf flood-list

To display a list of OSPF LSAs waiting to be flooded over an interface, use the **show ospf flood-list** command.

show ospf flood-list [**vrf** *name* | **all**] *interface_name*

Syntax Description

interface_name	The name of the interface for which to display neighbor information.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf flood-list** command:

```
> show ospf flood-list outside

  Interface outside, Queue length 20
  Link state flooding due in 12 msec

  Type  LS ID          ADV RTR          Seq NO       Age   Checksum
     5  10.2.195.0     192.168.0.163    0x80000009   0     0xFB61
     5  10.1.192.0     192.168.0.163    0x80000009   0     0x2938
     5  10.2.194.0     192.168.0.163    0x80000009   0     0x757
     5  10.1.193.0     192.168.0.163    0x80000009   0     0x1E42
     5  10.2.193.0     192.168.0.163    0x80000009   0     0x124D
     5  10.1.194.0     192.168.0.163    0x80000009   0     0x134C
```

show ospf interface

To display the OSPF-related interface information, use the **show ospf interface** command.

show ospf interface [**vrf** *name* | **all**] [*interface_name*]

Syntax Description

interface_name	(Optional) Name of the interface for which to display the OSPF-related information.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command Default

When you do not specify an interface name, the OSPF information for all interfaces is shown.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf interface** command:

```
> show ospf interface outside
out is up, line protocol is up
  Internet Address 10.0.3.4 mask 255.255.255.0, Area 0
  Process ID 2, Router ID 10.0.3.4, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State WAITING, Priority 1
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured, Hello 10 msec, Dead 1, Wait 1, Retransmit 5
    Hello due in 5 msec
    Wait time before Designated router selection 0:00:11
  Index 1/1, flood queue length 0
  Next 0x00000000(0)/0x00000000(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
```

show ospf neighbor

To display the OSPF-neighbor information on a per-interface basis, use the **show ospf neighbor** command.

show ospf neighbor [**vrf** *name* | **all**] [**detail** | *interface_name* [*nbr_router_id*]]

Syntax Description

detail	(Optional) Lists detail information for the specified router.
interface_name	(Optional) Name of the interface for which to display neighbor information.
nbr_router_id	(Optional) Router ID of the neighbor router.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf neighbor** command. It shows how to display the OSPF-neighbor information on a per-interface basis.

```
> show ospf neighbor outside

Neighbor 192.168.5.2, interface address 10.225.200.28
    In the area 0 via interface outside
    Neighbor priority is 1, State is FULL, 6 state changes
    DR is 10.225.200.28 BDR is 10.225.200.30
    Options is 0x42
    Dead timer due in 00:00:36
    Neighbor is up for 00:09:46
  Index 1/1, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

The following is sample output from the **show ospf neighbor detail** command. It shows how to display the detailed information for the specified OSPF-neighbor.

```
> show ospf neighbor detail

Neighbor 25.1.1.60, interface address 15.1.1.60
    In the area 0 via interface inside
    Neighbor priority is 1, State is FULL, 46 state changes
    DR is 15.1.1.62 BDR is 15.1.1.60
```

```
Options is 0x12 in Hello (E-bit, L-bit)
Options is 0x52 in DBD (E-bit, L-bit, O-bit)
LLS Options is 0x1 (LR), last OOB-Resync 00:03:07 ago
Dead timer due in 0:00:24
Neighbor is up for 01:42:15
Index 5/5, retransmission queue length 0, number of retransmission 0
First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
Last retransmission scan length is 0, maximum is 0
Last retransmission scan time is 0 msec, maximum is 0 msec
```

show ospf nsf

To display the OSPFv2 related NSF information, use the **show ospf nsf** command.

show ospf nsf [**vrf** *name* | **all**]

Syntax Description

[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf nsf** command:

```
> show ospf nsf
Routing Process "ospf 10"
Non-Stop Forwarding enabled
    Clustering is not configured in spanned etherchannel mode
IETF NSF helper support enabled
Cisco NSF helper support enabled
    OSPF restart state is
    Handle 1, Router ID 25.1.1.60, checkpoint Router ID 0.0.0.0
    Config wait timer interval 10, timer not running
    Dbase wait timer interval 120, timer not running
```

show ospf request-list

To display a list of all LSAs that are requested by a router, use the **show ospf request-list** command.

show ospf request-list [**vrf** *name* | **all**] *nbr_router_id interface_name*

Syntax Description

interface_name	Name of the interface for which to display neighbor information. Displays the list of all LSAs that are requested by the router from this interface.
nbr_router_id	Router ID of the neighbor router. Displays the list of all LSAs that are requested by the router from this neighbor.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf request-list** command:

```
> show ospf request-list 192.168.1.12 inside

         OSPF Router with ID (192.168.1.11) (Process ID 1)

  Neighbor 192.168.1.12, interface inside address 172.16.1.12

  Type   LS ID          ADV RTR        Seq NO       Age   Checksum
     1   192.168.1.12   192.168.1.12   0x8000020D   8     0x6572
```

Related Commands

Command	Description
show ospf retransmission-list	Displays a list of all LSAs waiting to be resent.

show ospf retransmission-list

To display a list of all LSAs waiting to be resent for a specific neighbor and interface, use the **show ospf retransmission-list** command.

show ospf retransmission-list [**vrf** *name* | **all**] *nbr_router_id interface_name*

Syntax Description

interface_name	Name of the interface for which to display neighbor information.
nbr_router_id	Router ID of the neighbor router.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf retransmission-list** command for the 192.168.1.11 neighbor router on the outside interface.

```
> show ospf retransmission-list 192.168.1.11 outside

         OSPF Router with ID (192.168.1.12) (Process ID 1)

  Neighbor 192.168.1.11, interface outside address 172.16.1.11

  Link state retransmission due in 3764 msec, Queue length 2
  Type   LS ID          ADV RTR        Seq NO       Age   Checksum
     1   192.168.1.12   192.168.1.12   0x80000210   0     0xB196
```

Related Commands

Command	Description
show ospf request-list	Displays a list of all LSAs that are requested by a router.

show ospf rib

To display the OSPF Router Information Base (RIB), use the **show ospf rib** command

show ospf [**vrf** *name* | **all**] [*process_id* [*area_id*]] **rib** [*network_prefix* [*network_mask*]] | **detail** | **redistribution** [*network_prefix* [*network_mask*]] | **detail**]]

Syntax Description

process_id	(Optional) The ID of the OSPF process.
area_id	(Optional) ID of the area that is associated with the OSPF address range.
network_prefix [*network_mask*]	(Optional) The network prefix and optionally the mask of the route you want to view, for example: 10.100.10.1 10.100.10.0 255.255.255.0
detail	(Optional) Display detailed information about the RIB.
redistribution	(Optional) Display redistribution information. You can also specify the network prefix and mask or **detail** keyword after the redistribution keyword.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

show ospf statistics

To display various OSPF statistics, such as the number of times SPF was executed, the reasons, and the duration, use the **show ospf statistics** command.

show ospf [**vrf** *name* | **all**] [*process_id*] **statistics** [**detail**]

Syntax Description

detail	(Optional) Specifies detailed SPF information, including the trigger points.
process_id	(Optional) Specifies an internal ID that is locally assigned and can be any positive integer. This ID is the number assigned administratively when the OSPF routing process is enabled.
[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf statistics** command:

```
> show ospf 10 statistics detail
Area 10: SPF algorithm executed 6 times

SPF 1 executed 04:36:56 ago, SPF type Full
  SPF calculation time (in msec):
  SPT    Prefix D-Int  Sum    D-Sum  Ext    D-Ext  Total
       0      0      0      0      0      0      0 0
  RIB manipulation time (in msec):
  RIB Update    RIB Delete
                0             0
  LSIDs processed R:1 N:0 Prefix:0 SN:0 SA:0 X7:0
  Change record R L
  LSAs changed 2
  Changed LSAs. Recorded is Advertising Router, LSID and LS type:
  49.100.168.192/0(R) 49.100.168.192/2(L)

SPF 2 executed 04:35:50 ago, SPF type Full
  SPF calculation time (in msec):
  SPT    Prefix D-Int  Sum    D-Sum  Ext    D-Ext  Total
       0      0      0      0      0      0      0 0
  RIB manipulation time (in msec):
  RIB Update    RIB Delete
                0             0
  LSIDs processed R:2 N:1 Prefix:0 SN:0 SA:0 X7:0
```

```
Change record R N L
LSAs changed 5
Changed LSAs. Recorded is Advertising Router, LSID and LS type:
50.100.168.192/0(R) 50.100.168.192/2(L) 49.100.168.192/0(R) 50.100.168.192/0(R)
50.100.168.192/2(N)
```

show ospf summary-address

To display a list of all summary address redistribution information that is configured under an OSPF process, use the **show ospf summary-address** command.

show ospf summary-address [**vrf** *name* | **all**]

Syntax Description

[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following shows sample output from the **show ospf summary-address** command. It shows how to display a list of all summary address redistribution information before a summary address has been configured for an OSPF process with the ID of 5.

```
> show ospf 5 summary-address

OSPF Process 2, Summary-address

10.2.0.0/255.255.0.0 Metric -1, Type 0, Tag 0
10.2.0.0/255.255.0.0 Metric -1, Type 0, Tag 10
```

show ospf traffic

To display a list of different types of packets that have been processed (sent or received) by a particular OSPF instance, use the **show ospf traffic** command.

show ospf traffic [**vrf** *name* | **all**]

Syntax Description

[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Usage Guidelines

With this command, you can get a snapshot of the different types of OSPF packets that are being processed without enabling debugging. If there are two OSPF instances configured, the **show ospf traffic** command displays the statistics for both instances with the process ID of each instance. You can also display the statistics for a single instance by using the **show ospf** *process_id* **traffic** command.

Examples

The following shows sample output from the **show ospf traffic** command.

```
> show ospf traffic

OSPF statistics (Process ID 70):

       Rcvd: 244 total, 0 checksum errors
             234 hello, 4 database desc, 1 link state req
             3 link state updates, 2 link state acks
       Sent: 485 total
             472 hello, 7 database desc, 1 link state req
             3 link state updates, 2 link state acks
```

Related Commands

Command	Description
show ospf virtual-links	Displays the parameters and the current state of OSPF virtual links.

show ospf virtual-links

To display the parameters and the current state of OSPF virtual links, use the **show ospf virtual-links** command.

show ospf virtual-links [**vrf** *name* | **all**]

Syntax Description

[**vrf** *name* | **all**]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. If you want the command to affect all virtual routers, include the **all** keyword. If you include neither of these VRF-related keywords, the command applies to the global VRF virtual router.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The [**vrf** *name* | **all**] keywords were added.

Examples

The following is sample output from the **show ospf virtual-links** command:

```
> show ospf virtual-links

Virtual Link to router 192.168.101.2 is up
Transit area 0.0.0.1, via interface Ethernet0, Cost of using 10
Transmit Delay is 1 sec, State POINT_TO_POINT
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 0:00:08
Adjacency State FULL
```