

# set logout

To set the number of minutes until the system disconnects an idle session automatically, use the **set logout** command.

**set logout** *timeout*

---

**Syntax Description**

<i>timeout</i>	Number of minutes until the system disconnects an idle session automatically; valid values are from 0 to 10,000 minutes.
----------------	--

---

---

**Defaults**

The default is 20 minutes.

---

**Command Types**

Switch command.

---

**Command Modes**

Privileged.

---

**Usage Guidelines**

Setting the value to 0 disables the automatic disconnection of idle sessions.

The **show tech-support** command may time out if the configuration file output takes longer to display than the configured session timeout time. If this happens, enter a **set logout** *timeout* value of 0 to disable automatic disconnection of idle sessions or enter a longer *timeout* value.

---

**Examples**

This example shows how to set the number of minutes until the system disconnects an idle session automatically:

```
Console> (enable) set logout 20
Sessions will be automatically logged out after 20 minutes of idle time.
Console> (enable)
```

This example shows how to disable the automatic disconnection of idle sessions:

```
Console> (enable) set logout 0
Sessions will not be automatically logged out.
Console> (enable)
```

---

**Related Commands**

[show tech-support](#)

## set mls agingtime

To specify the MLS aging time of shortcuts to an MLS entry in the Catalyst 6500 series switches, use the **set mls agingtime** command.

```
set mls agingtime [ip | ipx] {agingtime}
```

```
set mls agingtime fast {fastagingtime} {pkt_threshold}
```

```
set mls agingtime long-duration {longagingtime}
```

Syntax Description		
<b>ip</b>	(Optional) Specifies IP MLS.	
<b>ipx</b>	(Optional) Specifies IPX MLS.	
<i>agingtime</i>	MLS aging time of shortcuts to an MLS entry; valid values are multiples of 8 to any value in the range of 8 to 2024 seconds.	
<b>fast</b>	Specifies the MLS aging time of shortcuts to an MLS entry that has no more than <i>pkt_threshold</i> packets switched within <i>fastagingtime</i> seconds after it is created.	
<i>fastagingtime</i>	MLS aging time of shortcuts to an MLS entry; valid values are multiples of 8 to any value in the range from 0 to 128 seconds.	
<i>pkt_threshold</i>	Packet threshold value; valid values are <b>0, 1, 3, 7, 15, 31, 63,</b> and <b>127</b> packets.	
<b>long-duration</b>	Sets the aging time for active flows.	
<i>longagingtime</i>	MLS aging time of shortcuts to an MLS entry; valid values are 64 to 1920 seconds in increments of 64.	

**Defaults** The default *agingtime* is 256 seconds. The default *fastagingtime* is 0, no fast aging. The default *pkt\_threshold* is 0. The default *longagingtime* is 1920.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** If you use the **ip** keyword, you are specifying a shortcut for IP MLS. If you use the **ipx** keyword, you are specifying a shortcut for IPX MLS.

If you enter **0** for the *fastagingtime* value, fast aging is disabled.

If you do not specify *fastagingtime* or *pkt\_threshold*, the default value is used.

If you enter any of the **set mls** commands on a Catalyst 6500 series switch without MLS, this warning message displays:

```
MLS not supported on feature card.
```

The *agingtime* value can be configured as multiples of 8 in the range of 8 to 2024 seconds. The values are picked up in numerical order to achieve efficient aging. Any value for *agingtime* that is not a multiple of 8 seconds is adjusted to the closest one. For example, 65 is adjusted to 64, while 127 is adjusted to 128.

The *fastagingtime* value can be configured as multiples of 8 to any value in the range of 0 to 128 seconds.

The default *pkt\_threshold* value is 0. It can be configured as 0, 1, 3, 7, 15, 31, 63, or 127 (the values picked for efficient aging). If you do not configure *fastagingtime* exactly the same for these values, it adjusts to the closest value. A typical value for *fastagingtime* and *pkt\_threshold* is 32 seconds and 0 packet, respectively. (It means no packet switched within 32 seconds after the entry was created.)

The *agingtime* value applies to an MLS entry that has no more than *pkt\_threshold* packets switched within *fastagingtime* seconds after it is created. A typical example is the MLS entry destined to/sourced from a DNS or TFTP server. This entry may never be used again once it is created. For example, only one request goes to a server and one reply returns from the server, and then the connection is closed.

The **agingtime fast** option is used to purge entries associated with very short flows, such as DNS and TFTP.

Keep the number of MLS entries in the MLS cache below 32,000. If the number of MLS entries exceed 32,000, some flows (less than 1 percent) are sent to the router.

To keep the number of MLS cache entries below 32,000, decrease the aging time up to 8 seconds. If your switch has a lot of short flows used by only a few packets, then you can use fast aging.

If cache entries continue to exceed 32,000, decrease the normal aging time in 64-second increments from the 256-second default.

You can force an active flow to age out by entering the **set mls agingtime long-duration** command. You can specify the aging time of the active flow in the range of 64 to 1920 seconds in increments of 64.

---

## Examples

These examples show how to set the aging time:

```
Console> (enable) set mls agingtime 512  
IP Multilayer switching aging time set to 512 seconds.  
Console> (enable)
```

```
Console> (enable) set mls agingtime ipx 512  
IPX Multilayer switching aging time set to 512  
Console> (enable)
```

This example shows how to set the fast aging time:

```
Console> (enable) set mls agingtime fast 32 0  
Multilayer switching fast aging time set to 32 seconds for entries with no more than 0  
packet switched.  
Console> (enable)
```

This example shows how to set the aging time for active flows:

```
Console> (enable) set mls agingtime long-duration 128  
Multilayer switching agingtime set to 128 seconds for long duration flows  
Console> (enable)
```

■ set mls agingtime

**Related Commands** [clear mls statistics entry](#)  
[show mls](#)

# set mls bridged-flow-statistics

To enable or disable statistics for bridged flows for specified VLANs, use the **set mls bridged-flow-statistics** command.

```
set mls bridged-flow-statistics { enable | disable } { vlanlist }
```

Syntax Description	enable	Enables statistics for bridged flows.
	disable	Disables statistics for bridged flows
	<i>vlanlist</i>	Number of the VLAN or VLANs; valid values are 1 to 1000, 1025 to 4094. See the “Usage Guidelines” section for more information.

**Defaults** By default, bridged-flow statistics is disabled on all VLANs.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** You can enter one or multiple VLANs. The following examples are valid VLAN lists: **1; 1,2,3; 1-3,7**. Bridged flows are exported through NDE when bridged flow statistics is enabled.

**Examples** This example shows how to enable bridged-flow statistics on the specified VLANs:

```
Console> (enable) set mls bridged-flow-statistics enable 1-21
Netflow statistics is enabled for bridged packets on vlan(s) 1-21.
Console> (enable)
```

**Related Commands**

- [show mls nde](#)
- [show mls entry](#)
- [show mls statistics](#)

# set mls cef load-balance

To include or exclude Layer 4 ports in a load-balancing hash, use the **set mls cef load-balance** command.

**set mls cef load-balance {full | source-destination-ip}**

Syntax Description	full	Bases the hash on Layer 4 ports and source and destination IP addresses.
	<b>source-destination-ip</b>	Bases the hash on source and destination IP addresses.

**Defaults** By default, the load-balancing hash is based on source and destination IP addresses.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** When multiple paths are available to reach a destination, the new hash is used to choose the path to be used for forwarding.

**Examples** This example shows how to base the hash on Layer 4 ports and source and destination IP addresses:

```
Console> (enable) set mls cef load-balance full
Console> (enable)
```

This example shows how to base the hash on source and destination IP addresses:

```
Console> (enable) set mls cef load-balance source-destination-ip
Console> (enable)
```

**Related Commands** [show mls](#)

# set mls cef per-prefix-statistics

To set MLS CEF per-prefix statistics mode, use the **set mls cef per-prefix statistics** command.

```
set mls cef per-prefix statistics {enable | disable}
```

Syntax Description	enable	Disables per-prefix statistics for all FIB entries
	disable	Disables per-prefix statistics for all FIB entries.

**Defaults** MLS CEF per-prefix statistics mode is enabled.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** When the **set mls cef per-prefix-statistics** command is enabled, the switch makes a best effort to allocate adjacencies with statistics for each prefix. Statistics for a prefix are computed by adding up the packet/byte counts of all the adjacencies that are associated with the prefix. Because only half of the adjacency table entries have statistics, all prefixes might not be associated with adjacencies with statistics.

**Examples** This example shows how to enable per-prefix statistics for all FIB entries:

```
Console> (enable) set mls cef per-prefix-stats enable
Per prefix stats is enabled
Console> (enable)
```

This examples shows how to disable per-prefix statistics for all FIB entries:

```
Console> (enable) set mls cef per-prefix-stats disable
Per prefix stats is disabled
Console> (enable)
```

**Related Commands** [show mls](#)

# set mls exclude protocol

To exclude an MLS protocol port on a switch configured with the Supervisor Engine 1 with Layer 3 Switching Engine WS-F6K-PFC, use the **set mls exclude protocol** command. To exclude protocols from statistics gathering on switches configured with the Supervisor Engine 2 with Layer 3 Switching Engine II (PFC2), use the **set mls exclude protocol** command.

```
set mls exclude protocol {tcp | udp | both} {port_number | port_name}
```

Syntax Description	
<b>tcp   udp   both</b>	Specifies a TCP, UDP port, or that the port be applied to both TCP and UDP traffic.
<i>port_number</i>	Number of the protocol port; valid values are from 1 to 65535.
<i>port_name</i>	Name of the port; valid values are <b>dns</b> , <b>ftp</b> , <b>smtp</b> , <b>telnet</b> , <b>x</b> , <b>www</b> .

**Defaults** This command has no default settings.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** If you enter any of the **set mls** commands on a Catalyst 6500 series switch without MLS, this warning message is displayed:

```
MLS not supported on feature card.
```

You can add a maximum of four protocol ports to the exclude table.

MLS exclusion is supported in full flow mode only.

If you enter **x** for the port name, this specifies the Layer 4 port used by the X-windows application.

**Examples** This example shows how to exclude TCP packets on protocol port 6017:

```
Console> (enable) set mls exclude protocol tcp 6017
TCP packets with protocol port 6017 will be switched by RP.
Console> (enable)
```

This example shows how to exclude UDP packets on protocol port 6017:

```
Console> (enable) set mls exclude protocol udp 6017
TCP and UDP packets with protocol port 6017 will be switched by RP.
Console> (enable)
```

**Related Commands** [show mls](#)

# set mls flow

To specify the minimum flow mask used for MLS, use the **set mls flow** command. This command is needed to collect statistics for the supervisor engine.

```
set mls flow { destination | destination-source | full }
```



### Caution

Use this command carefully. This command *purges all existing shortcuts* and affects the number of active shortcuts. This command can increase the cache usage and increase the load on the router.



### Caution

Be extremely careful if you enter this command on a switch that already has a large number of shortcuts (greater than 16,000).



### Caution

Do not place this command in scripts that are frequently executed—changing the MLS flow mask purges all MLS cache entries.

### Syntax Description

<b>destination</b>	Sets the minimum flow mask to destination flow.
<b>destination-source</b>	Sets the minimum flow mask to source flow.
<b>full</b>	Sets the minimum flow mask to an extended access list.

### Defaults

If there are no access lists on any MLS-RP, the flow mask is set to destination flow.

### Command Types

Switch command.

### Command Modes

Privileged.

### Usage Guidelines

This command specifies the minimum MLS flow mask. Depending on the MLS-RP configuration, the actual flow mask used might be more specific than the specified minimum flow mask. For example, if you configure the minimum flow mask to **destination-source**, but an MLS-RP interface is configured with IP extended access lists, the actual flow mask used will be **full**.

If you configure a more specific flow mask (for example, **destination-source** or **full**), the number of active flow entries increases. To limit the number of active flow entries, you might need to decrease the MLS aging time.

This command is intended to be used for gathering very detailed statistics at the protocol port level—for example, when NetFlow data is exported to an RMON2 probe.

---

**Examples**

These examples show how to specify that only expired flows to subnet 171.69.194.0 are exported:

```
Console> (enable) set mls flow destination  
Configured flow mask is set to destination flow.  
Console> (enable)
```

```
Console> (enable) set mls flow destination-source  
Configured flow mask is set to destination-source flow.  
Console> (enable)
```

```
Console> (enable) set mls flow full  
Configured flow mask is set to full flow.  
Console> (enable)
```

---

**Related Commands**    [show mls](#)

## set mls nde

To configure the NetFlow Data Export (NDE) feature in the Catalyst 6500 series switches to allow command-exporting statistics to be sent to the preconfigured collector, use the **set mls nde** command.

```
set mls nde {enable | disable}
```

```
set mls nde {collector_ip | collector_name} {udp_port_num}
```

```
set mls nde version {1 | 5 | 7 | 8}
```

```
set mls nde flow [exclude | include] [destination ip_addr_spec] [source ip_addr_spec]
[protocol protocol] [src-port src_port] [dst-port dst_port]
```

```
set mls nde {destination-ifindex | source-ifindex} {enable | disable}
```

### Syntax Description

<b>enable</b>	Enables NDE.
<b>disable</b>	Disables NDE.
<i>collector_ip</i>	IP address of the collector if DNS is enabled.
<i>collector_name</i>	Name of the collector if DNS is enabled.
<i>udp_port_num</i>	Number of the UDP port to receive the exported statistics.
<b>version</b>	Specifies the version of the NDE; valid versions are <b>1</b> , <b>5</b> , <b>7</b> , and <b>8</b> .
<b>1   5   7   8</b>	Version of the NDE feature.
<b>flow</b>	Adds filtering to NDE.
<b>exclude</b>	(Optional) Allows exporting of all flows except the flows matching the given filter.
<b>include</b>	(Optional) Allows exporting of all flows matching the given filter.
<b>destination</b>	(Optional) Specifies the destination IP address.
<i>ip_addr_spec</i>	(Optional) Full IP address or a subnet address in these formats: <i>ip_addr</i> , <i>ip_addr/netmask</i> , or <i>ip_addr/maskbit</i> .
<b>source</b>	(Optional) Specifies the source IP address.
<b>protocol</b>	(Optional) Specifies the protocol type.
<i>protocol</i>	(Optional) Protocol type; valid values can be a number from 0 to 255 or <b>ip</b> , <b>ipinip</b> , <b>icmp</b> , <b>igmp</b> , <b>tcp</b> , or <b>udp</b> . <b>0</b> indicates “do not care.”
<b>src-port</b> <i>src_port</i>	(Optional) Specifies the number of the TCP/UDP source port (decimal). Used with <b>dst-port</b> to specify the port pair if the <b>protocol</b> is <b>tcp</b> or <b>udp</b> . <b>0</b> indicates “do not care.”
<b>dst-port</b> <i>dst_port</i>	(Optional) Specifies the number of the TCP/UDP destination port (decimal). Used with <b>src-port</b> to specify the port pair if the <b>protocol</b> is <b>tcp</b> or <b>udp</b> . <b>0</b> indicates “do not care.”
<b>destination-ifindex</b>	Specifies destination ifIndex support.
<b>source-ifindex</b>	Specifies source ifIndex support.
<b>enable</b>	Enables ifIndex support.
<b>disable</b>	Disables ifIndex support.

**Defaults**

The defaults are Netflow Data Export version 7, and all expired flows are exported until the filter is specified explicitly. Destination ifIndex support and source ifIndex support are enabled.

**Command Types**

Switch command.

**Command Modes**

Privileged.

**Usage Guidelines**

If you enter any **set mls nde** commands on a Catalyst 6500 series switch without MLS, this warning message is displayed:

```
mls not supported on feature card.
```

When you try to enable NDE and there are previously configured filtered flows on the switch, this warning message is displayed:

```
Console> (enable) set mls nde enable
Netflow export configured for port 80 on host 172.20.25.101
Netflow export enabled.
Warning!! There is a potential statistics mismatch due to existing excluded
protocols.
```

When you try to add a filter to exclude some protocol packets and NDE is currently enabled, this warning message is displayed:

```
Console> (enable) set mls exclude protocol tcp 80
Netflow tables will not create entries for TCP packets with protocol port
80.
Warning!! There's a potential statistics mismatch due to enabled NDE.
```

Before you use the **set mls nde** command for the first time, you must configure the host to collect MLS statistics. The host name and UDP port number are saved in NVRAM, so you do not need to specify them. If you specify a host name and UDP port, values in NVRAM overwrite the old values. Collector values in NVRAM do not clear when NDE is disabled because this command configures the collector but does not enable NDE automatically.

The **set mls nde enable** command enables NDE, exporting statistics to the preconfigured collector.

If the *protocol* is not **tcp** or **udp**, set the **dst-port** *dst\_port* and **src-port** *src\_port* values to 0; otherwise, no flows are displayed.

If you try to enable NDE without first specifying a collector, you see this display:

```
Console> (enable) set mls nde enable
Please set host name and UDP port number with 'set mls nde <collector_name | collector_ip>
<udp_port_number>'.
Console> (enable)
```

The **set mls nde flow** command adds filtering to the NDE. Expired flows matching the specified criteria are exported. These values are stored in NVRAM and do not clear when NDE is disabled. If any option is not specified in this command, it is treated as a wildcard. The NDE filter in NVRAM does not clear when NDE is disabled.

Only one filter can be active at a time. If you do not enter the **exclude** or **include** keyword, the filter is assumed to be an inclusion filter.

Use the following syntax to specify an IP subnet address:

- *ip\_subnet\_addr*—This is the short subnet address format. The trailing decimal number 00 in an IP address YY.YY.YY.00 specifies the boundary for an IP subnet address. For example, 172.22.36.00 indicates a 24-bit subnet address (subnet mask 172.22.36.00/255.255.255.0), and 173.24.00.00 indicates a 16-bit subnet address (subnet mask 173.24.00.00/255.255.0.0). However, this format can identify only a subnet address of 8, 16, or 24 bits.
- *ip\_addr/subnet\_mask*—This is the long subnet address format. For example, 172.22.252.00/255.255.252.00 indicates a 22-bit subnet address. This format can specify a subnet address of any bit number. To provide more flexibility, the *ip\_addr* is a full host address, such as 172.22.253.1/255.255.252.00.
- *ip\_addr/maskbits*—This is the simplified long subnet address format. The mask bits specify the number of bits of the network masks. For example, 172.22.252.00/22 indicates a 22-bit subnet address. The *ip\_addr* is a full host address, such as 193.22.253.1/22, which has the same subnet address as the *ip\_subnet\_addr*.

When you use the **set mls nde {collector\_ip | collector\_name} {udp\_port\_num}** command, the host name and UDP port number are saved in NVRAM and need not be specified again. If you specify a host name and UDP port, the new values overwrite the values in NVRAM. Collector values in NVRAM do not clear when you disable NDE.

## Examples

This example shows how to specify that only expired flows to a specific subnet are exported:

```
Console> (enable) set mls nde flow include destination 171.69.194.140/24
NDE destination filter set to 171.69.194.0/24
Console> (enable)
```

This example shows how to specify that only expired flows to a specific host are exported:

```
Console> (enable) set mls nde flow include destination 171.69.194.140
NDE destination filter set to 171.69.194.140/32.
Console> (enable)
```

This example shows how to specify that only expired flows from a specific subnet to a specific host are exported:

```
Console> (enable) set mls nde flow include destination 171.69.194.140/24 source 171.69.173.5/24
NDE destination filter set to 171.69.194.0/24, source filter set to 171.69.173.0/24
Console> (enable)
```

This example shows how to specify that only flows from a specific port are exported:

```
Console> (enable) set mls nde flow include dst_port 23
NDE source port filter set to 23.
Console> (enable)
```

This example shows how to specify that only expired flows from a specific host that are of a specified protocol are exported:

```
Console> (enable) set mls nde flow include source 171.69.194.140 protocol 51
NDE destination filter set to 171.69.194.140/32, protocol set to 51.
Console> (enable)
```

This example shows how to specify that all expired flows except those from a specific host to a specific destination port are exported:

```
Console> (enable) set mls nde flow exclude source 171.69.194.140 dst_port 23  
NDE destination filter set to 171.69.194.140/32, source port filter set to 23.  
Flows matching the filter will be excluded.  
Console> (enable)
```

This example shows how to disable destination ifIndex support:

```
Console> (enable) set mls nde destination-ifindex disable  
destination-index export has been disabled.  
Console> (enable)
```

This example shows how to disable source ifIndex support:

```
Console> (enable) set mls nde source-ifindex disable  
source-index export has been disabled.  
Console> (enable)
```

---

**Related Commands**

[clear mls nde flow](#)  
[show mls](#)  
[show mls nde](#)

# set mls rate

To set the rate at which index-directed packets are sent to the MSFC, use the **set mls rate** command.

```
set mls rate kpps
```

---

<b>Syntax Description</b>	<i>kpps</i>	MLS rate in thousands of packets per second; valid values are from 0 to 700. See the “Usage Guidelines” section for more information.
---------------------------	-------------	---

---

---

<b>Defaults</b>	The <i>kpps</i> argument is 0.
-----------------	--------------------------------

---

<b>Command Types</b>	Switch command.
----------------------	-----------------

---

<b>Command Modes</b>	Privileged.
----------------------	-------------

---

<b>Usage Guidelines</b>	You disable MLS rate limiting when you set the <i>kpps</i> argument to 0. When you disable MLS rate limiting, the switch bridges packets to the MSFC; packets are not index-directed.
-------------------------	---

---

<b>Examples</b>	This example shows how to set MLS rate limiting to 100 kpps:
-----------------	--

```
Console> (enable) set mls rate 100  
MLS rate limiting set to 100 Kpps  
Console> (enable)
```

This example shows how to disable MLS rate limiting:

```
Console> (enable) set mls rate 0  
MLS rate limiting disabled  
Console> (enable)
```

---

<b>Related Commands</b>	<a href="#">show mls</a>
-------------------------	--------------------------

# set mls statistics protocol

To add protocols to the protocols statistics list, use the **set mls statistics protocol** command.

```
set mls statistics protocol protocol src_port
```

<b>Syntax Description</b>	<i>protocol</i>	Name or number of the protocol; valid values are from 1 to 255, <b>ip</b> , <b>ipinip</b> , <b>icmp</b> , <b>igmp</b> , <b>tcp</b> , and <b>udp</b> .
	<i>src_port</i>	Number or type of the source port; valid values are from 1 to 65535, <b>dns</b> , <b>ftp</b> , <b>smtp</b> , <b>telnet</b> , <b>x</b> , and <b>www</b> .

**Defaults** This command has no default settings.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** If you enter any **set mls** commands on a Catalyst 6500 series switch without MLS, this warning message is displayed:

```
MLS not supported on feature card.
```

You can configure a maximum of 64 ports using the **set mls statistics protocol** command.

If you enter **x** for the source port, this specifies the Layer 4 port used by the X-windows application.

**Examples** This example shows how to set protocols for statistic collection:

```
Console> (enable) set mls statistics protocol 17 1934
Protocol 17 port 1934 is added to protocol statistics list.
Console> (enable)
```

**Related Commands** [clear mls statistics entry](#)  
[show mls statistics](#)

# set mls verify

To enable or disable checksum or packet checking based on packet length, use the **set mls verify** command.

```
set mls verify checksum {enable | disable}
```

```
set mls verify length {ip | ipx | both} {minimum | inconsistent} {enable | disable}
```

Syntax Description	checksum	Specifies IP checksum.
	enable	Enables IP checksum.
	disable	Disables IP checksum.
	length	Specifies checking IP or IPX packets based on packet length.
	ip   ipx   both	Specifies the type of packet.
	minimum	Specifies checking minimum packet length.
	inconsistent	Specifies checking inconsistent packet length. See the “Usage Guidelines” section for more information.
	enable	Enables checking IP or IPX packets based on packet length.
	disable	Disables checking IP or IPX packets based on packet length.

## Defaults

IP checksum is enabled.

Checking IP and IPX packets based on minimum and inconsistent packet length is enabled.

## Command Types

Switch command.

## Command Modes

Privileged.

## Usage Guidelines

The **set mls verify** command is available on Supervisor Engine 2 (WS-X6K-SUP2-2GE).

If you enable IP checksum or packet checking based on packet length, the Layer 3 ASIC drops the Layer 3 error packets that it encounters. If you disable this feature, the packets are not dropped.



**Note** We recommend that you do not disable IP checksum or packet checking based on packet length unless you have a specific need to pass non-standard packets.

Checking for inconsistent packet length means that the switch checks for an inconsistency between the physical length of the packet and the length coded in the packet.

---

**Examples**

This example shows how to enable IP checksum:

```
Console> (enable) set mls verify checksum enable  
Ip checksum verification enabled  
Console> (enable)
```

This example shows how to enable checking inconsistent IP and IPX packet length:

```
Console> (enable) set mls verify length both inconsistant enable  
Ip inconsistant length verification enabled  
Ip inconsistant length verification enabled  
Console> (enable)
```

This example shows how to disable checking minimum IPX packet length:

```
Console> (enable) set mls verify length ipx minimum disable  
Ipx minimum length verification disabled  
Console> (enable)
```

---

**Related Commands**

[show mls verify](#)

# set module

To enable or disable a module, use the **set module** command.

**set module enable | disable** *mod*

Syntax Description	enable	Enables a module.
	disable	Disables a module.
	<i>mod</i>	Number of the module.

**Defaults** The default is all modules are enabled.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** Avoid disabling a module when you are connected through a Telnet session; if you disable your session, you will disconnect your Telnet session.

If there are no other network connections to a Catalyst 6500 series switch (for example, on another module), you have to reenable the module from the console.

You can specify a series of modules by entering a comma between each module number (for example, 2,3,5). You can specify a range of modules by entering a dash between module numbers (for example, 2-5).

The **set module disable** command does not cut off the power to a module, it only disables the module. To turn off power to a module, refer to the **set module power** command.

If an individual port on a module was previously disabled, enabling the module does not enable the disabled port.

**Examples** This example shows how to enable module 2:

```
Console> (enable) set module enable 2
Module 2 enabled.
Console> (enable)
```

This example shows how to disable module 3 when connected through the console port:

```
Console> (enable) set module disable 3
Module 3 disabled.
Console> (enable)
```

This example shows how to disable module 2 when connected via a Telnet session:

```
Console> (enable) set module disable 2  
This command may disconnect your telnet session.  
Do you want to continue (y/n) [n]? y  
Module 2 disabled.  
Console> (enable)
```

**Related Commands**    [show module](#)

# set module name

To set the name for a module, use the **set module name** command.

```
set module name mod [mod_name]
```

<b>Syntax Description</b>	<i>mod</i>	Number of the module.
	<i>mod_name</i>	(Optional) Name created for the module.

**Defaults** The default is no module names are configured for any modules.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** If no module name is specified, any previously specified name is cleared.  
Use the **set module name** command to set the module for the MSM. Additional **set module** commands are not supported by the MSM.

**Examples** This example shows how to set the name for module 1 to Supervisor:

```
Console> (enable) set module name 1 Supervisor  
Module name set.  
Console> (enable)
```

**Related Commands** [show module](#)

# set module power

To turn the power on or off to a module, use the **set module power** command.

**set module power up | down** *mod*

Syntax Description	up	Turns on the power to a module.
	<b>down</b>	Turns off the power to a module.
	<i>mod</i>	Number of the module.

**Defaults** The default is power is on to a module.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** The **set module power up** command allows you to check if adequate power is available in the system to turn the power on. If not enough power is available, the module status changes from power-down to power-deny, and this message is displayed:

```
Module 4 could not be powered up due to insufficient power.
```

**Examples** This example shows how to power up module 4:

```
Console> (enable) set module power up 4
Module 4 powered up.
Console> (enable)
```

This example shows how to power down module 4:

```
Console> (enable) set module power down 4
Module 4 powered down.
Console> (enable)
```

**Related Commands** [show environment](#)

# set module shutdown

To shut down the NAM and Intrusion Detection System Module (IDS), use the **set module shutdown** command.

```
set module shutdown all | mod
```

<b>Syntax Description</b>	<b>all</b>	Shuts down NAM and IDSs.
	<b>mod</b>	Number of the module.

**Defaults** This command has no default settings.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** If you use the **set module shutdown** command, the configuration is not saved in NVRAM. The next time when the module boots up, it will come online. You can either reinsert or reset the module to bring it online.

If there are no other network connections to a Catalyst 6500 series switch (for example, on another module), you have to reenable the module from the console.

You can specify a series of modules by entering a comma between each module number (for example, 2,3,5).

**Examples** This example shows how to shutdown the NAM or IDS:

```
Console> (enable) set module shutdown 2
Console> (enable)
```

# set msfcautostate

To enable or disable the line protocol state determination of the Multilayer Switch Feature Cards (MSFCs) due to port state changes, use the **set msfcautostate** command.

**set msfcautostate {enable | disable}**

Syntax Description	enable	disable
	Activates the line protocol state determination.	Deactivates the line protocol state determination.

**Defaults** The default is enabled.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** This feature is used to accurately reflect the Layer 3 interface status based on the underlying Layer 2 interface status so that routing and other protocols converge faster. Faster protocol convergence prevents traffic from being discarded without notice.

When you enable the MSFC auto state feature, VLAN interfaces on the MSFC are active only when there is at least one other active interface in the spanning tree forwarding state on the Catalyst 6500 series switch. This interface could be a physical end-user port, a trunk connection for which the VLAN is active, or even another MSFC with an equivalent VLAN interface.

If you enable and then disable or disable and then enable the **set msfcautostate** command, you might have to use the **shutdown** and **no shutdown** commands to disable and then restart the VLAN and WAN interfaces on the MSFC.

If your FXS module ports are in an auxiliary VLAN and there are no switching module ports active in the VLAN, the FXS module will not initialize because the MSFC auto state feature shuts down all MSFC interfaces and subinterfaces. We recommend that you add a physical Ethernet port to the VLAN.



### Caution

You should not disable the MSFC auto state feature because the Layer 3 interface status might not accurately reflect the Layer 2 interface status. If you disable this feature, traffic might be discarded without notice even though other valid traffic paths might exist.

**Examples** This example shows how to enable the line protocol state determination of the MSFC:

```
Console> (enable) set msfcautostate enable
Console> (enable)
```

This example shows how to disable the line protocol state determination of the MSFC:

```
Console> (enable) set msfcautostate disable
Console> (enable)
```

**Related Commands** [show msfcautostate](#)

# set msmautostate

To enable or disable the line protocol state determination of the MSMs due to port state changes, use the **set msmautostate** command.

```
set msmautostate {enable | disable}
```

Syntax Description	enable	Deactivates the line protocol state determination.
	enable	Activates the line protocol state determination.

**Defaults** The default configuration has line protocol state determination disabled.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** This feature is useful for discontinuing the advertisement of routing paths when access to them is severed (either through fault or administrative disabling).

When you enable **msmautostate**, VLAN interfaces on the MSM are active only when there is at least one other active interface within the Catalyst 6500 series switch. This could be a physical end-user port, a trunk connection for which the VLAN is active, or even another MSM with an equivalent VLAN interface.

If you disable **msmautostate**, you might have to use the **shutdown** and **no shutdown** commands to disable and then restart the VLAN interface to bring the MSM back up.

**Examples** This example shows how to enable the line protocol state determination of the MSM:

```
Console> (enable) set msmautostate enable
MSM port auto state enabled.
Console> (enable)
```

This example shows how to disable the line protocol state determination of the MSM:

```
Console> (enable) set msmautostate disable
MSM port auto state disabled.
Console> (enable)
```

**Related Commands** [show msmautostate](#)

# set multicast router

To configure a port manually as a multicast router port, use the **set multicast router** command.

**set multicast router** *mod/port*

---

**Syntax Description**

*mod/port* Number of the module and port on the module.

---

---

**Defaults**

The default is no ports are configured as multicast router ports.

---

**Command Types**

Switch command.

---

**Command Modes**

Privileged.

---

**Usage Guidelines**

When you enable IGMP snooping, the ports to which a multicast-capable router is attached are identified automatically. The **set multicast router** command allows you to configure multicast router ports statically.

---

**Examples**

This example shows how to configure a multicast router port:

```
Console> (enable) set multicast router 3/1  
Port 3/1 added to multicast router port list.  
Console> (enable)
```

---

**Related Commands**

[clear multicast router](#)  
[set igmp](#)  
[show multicast group count](#)  
[show multicast router](#)

# set ntp broadcastclient

To enable or disable NTP in broadcast-client mode, use the **set ntp broadcastclient** command.

**set ntp broadcastclient {enable | disable}**

<b>Syntax Description</b>	<table border="1"> <tr> <td><b>enable</b></td> <td>Enables NTP in broadcast-client mode.</td> </tr> <tr> <td><b>disable</b></td> <td>Disables NTP in broadcast-client mode.</td> </tr> </table>	<b>enable</b>	Enables NTP in broadcast-client mode.	<b>disable</b>	Disables NTP in broadcast-client mode.
<b>enable</b>	Enables NTP in broadcast-client mode.				
<b>disable</b>	Disables NTP in broadcast-client mode.				
<b>Defaults</b>	The default is broadcast-client mode is disabled.				
<b>Command Types</b>	Switch command.				
<b>Command Modes</b>	Privileged.				
<b>Usage Guidelines</b>	The broadcast-client mode assumes that a broadcast server, such as a router, sends time-of-day information regularly to a Catalyst 6500 series switch.				
<b>Examples</b>	<p>This example shows how to enable an NTP broadcast client:</p> <pre>Console&gt; (enable) <b>set ntp broadcastclient enable</b> NTP Broadcast Client mode enabled. Console&gt; (enable)</pre> <p>This example shows how to disable an NTP broadcast client:</p> <pre>Console&gt; (enable) <b>set ntp broadcastclient disable</b> NTP Broadcast Client mode disabled. Console&gt; (enable)</pre>				
<b>Related Commands</b>	<a href="#">show ntp</a>				

# set ntp broadcastdelay

To configure a time-adjustment factor so the Catalyst 6500 series switch can receive broadcast packets, use the **set ntp broadcastdelay** command.

```
set ntp broadcastdelay microseconds
```

---

<b>Syntax Description</b>	<i>microseconds</i>	Estimated round-trip time, in microseconds, for NTP broadcasts; valid values are from 1 to 999999.
---------------------------	---------------------	--

---

---

<b>Defaults</b>	The default is the NTP broadcast delay is set to 3000 milliseconds.
-----------------	---

---

<b>Command Types</b>	Switch command.
----------------------	-----------------

---

<b>Command Modes</b>	Privileged.
----------------------	-------------

---

<b>Examples</b>	This example shows how to set the NTP broadcast delay to 4000 milliseconds:
-----------------	---

```
Console> (enable) set ntp broadcastdelay 4000  
NTP broadcast delay set to 4000 microseconds.  
Console> (enable)
```

---

<b>Related Commands</b>	<a href="#">show ntp</a>
-------------------------	--------------------------

# set ntp client

To enable or disable a Catalyst 6500 series switch as an NTP client, use the **set ntp client** command.

```
set ntp client { enable | disable }
```

Syntax Description	enable	disable
	Enables a Catalyst 6500 series switch as an NTP client.	Disables a Catalyst 6500 series switch as an NTP client.

**Defaults** The default is NTP client mode is disabled.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** You can configure NTP in either broadcast-client mode or client mode. The broadcast-client mode assumes that a broadcast server, such as a router, sends time-of-day information regularly to a Catalyst 6500 series switch. The client mode assumes that the client (a Catalyst 6500 series switch) regularly sends time-of-day requests to the NTP server.

**Examples** This example shows how to enable NTP client mode:

```
Console> (enable) set ntp client enable
NTP client mode enabled.
Console> (enable)
```

**Related Commands** [show ntp](#)

## set ntp server

To specify the NTP server address and configure an NTP server authentication key, use the **set ntp server** command.

```
set ntp server ip_addr [key public_keynum]
```

<b>Syntax Description</b>	<i>ip_addr</i>	IP address of the NTP server.
	<b>key</b> <i>public_keynum</i>	(Optional) Specifies the key number; valid values are 1 to 4292945295.

**Defaults** This command has no default settings.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** The client mode assumes that the client (a Catalyst 6500 series switch) sends time-of-day requests regularly to the NTP server. A maximum of ten servers per client is allowed.

**Examples** This example shows how to configure an NTP server:

```
Console> (enable) set ntp server 172.20.22.191
NTP server 172.20.22.191 added.
Console> (enable)
```

**Related Commands** [clear ntp server](#)  
[show ntp](#)

# set ntp summertime

To set the clock ahead one hour during daylight saving time, use the **set ntp summertime** command.

**set ntp summertime** {enable | disable} [*zone*]

**set ntp summertime recurring** [{*week*} {*day*} {*month*} {*hh:mm*} {*week* | *day* | *month* | *hh:mm*} [*offset*]]

**set ntp summertime date** {*month*} {*date*} {*year*} {*hh:mm*} {*month* | *date* | *year* | *hh:mm*} [*offset*]

## Syntax Description

<b>enable</b>	Causes the system to set the clock ahead one hour during daylight saving time.
<b>disable</b>	Prevents the system from setting the clock ahead one hour during daylight saving time.
<i>zone</i>	(Optional) Time zone used by the <b>set summertime</b> command.
<b>recurring</b>	Specifies the summertime dates that recur every year.
<i>week</i>	(Optional) Week of the month ( <b>first, second, third, fourth, last, 1...5</b> ).
<i>day</i>	(Optional) Day of the week ( <b>Sunday, Monday, Tuesday</b> , and so forth).
<i>month</i>	Month of the year ( <b>January, February, March</b> , and so forth).
<i>hh:mm</i>	Hours and minutes.
<i>offset</i>	(Optional) Amount of offset in minutes (1 to 1440 minutes).
<i>date</i>	Day of the month (1 to 31).
<i>year</i>	Number of the year (1993 to 2035).

## Defaults

By default, the **set ntp summertime** command is disabled. Once enabled, the default for *offset* is 60 minutes, following U.S. standards.

## Command Types

Switch command.

## Command Modes

Privileged.

## Usage Guidelines

After you enter the **clear config** command, the dates and times are set to default.

Unless you configure it otherwise, this command advances the clock one hour at 2:00 a.m. on the first Sunday in April and moves back the clock one hour at 2:00 a.m. on the last Sunday in October.

**Examples**

This example shows how to cause the system to set the clock ahead one hour during daylight saving time:

```
Console> (enable) set ntp summertime enable PDT
Summertime is enabled and set to "PDT".
Console> (enable)
```

This example shows how to prevent the system from setting the clock ahead one hour during daylight saving time:

```
Console> (enable) set ntp summertime disable
Summertime disabled.
Console> (enable)
```

This example shows how to set daylight saving time to the zonename AUS and repeat every year, starting from the third Monday of February at noon and ending at the second Saturday of August at 3:00 p.m. with an offset of 30 minutes:

```
Console> (enable) set ntp summertime AUS recurring 3 Mon Feb 12:00 2 Saturday Aug 15:00 30
Summer time is disabled and set to 'AUS' with offset 30 minutes.
  start: 12:00:00 Sun Feb 13 2000
  end:   14:00:00 Sat Aug 26 2000
  Recurring, starting at 12:00:00 on Sunday of the third week of February and ending
  on Saturday of the fourth week of August.
Console> (enable)
```

This example shows how to set the daylight saving time to start on January 29, 1999 at 2:00 a.m. and end on August 19, 2004 at 3:00 p.m. with an offset of 30 minutes:

```
Console> (enable) set ntp summertime date jan 29 1999 02:00 aug 19 2004 15:00 30
Summertime is disabled and set to ''
Start : Fri Jan 29 1999, 02:00:00
End   : Thu Aug 19 2004, 15:00:00
Offset: 30 minutes
Recurring: no
Console> (enable)
```

This example shows how to set recurring to reset default to US summertime:

```
Console> (enable) set ntp summertime recurring 3 mon feb 4 thurs oct 8:00 500
Command authorization none.
Summertime is enabled and set to ''
Start : Mon Feb 21 2000, 03:00:00
End   : Fri Oct 20 2000, 08:00:00
Offset: 500 minutes (8 hours 20 minutes)
Recurring: yes, starting at 03:00am of third Monday of February and ending on 08:00am of
fourth Thursday of October.
Console> (enable)
```

**Related Commands** [show ntp](#)

# set ntp timezone

To configure the time offset from Greenwich Mean Time, use the **set ntp timezone** command.

```
set timezone [zone_name] [hours [minutes]]
```

Syntax Description	
<i>zone_name</i>	(Optional) Name of the time zone.
<i>hours</i>	(Optional) Time offset (hours) from Greenwich Mean Time; valid values are from -12 to 12 hours.
<i>minutes</i>	(Optional) Time offset (minutes) from Greenwich Mean Time; valid values are 0 to 59 minutes.

**Defaults** This command has no default settings.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** The **set ntp timezone** command is effective only when NTP is running. If you set the time explicitly and NTP is disengaged, the **set ntp timezone** command has no effect. If you have enabled NTP and have not entered the **set timezone** command, the Catalyst 6500 series switch displays UTC by default.

**Examples** This example shows how to set the time zone to Pacific Standard Time with an offset of minus 8 hours from UTC:

```
Console> (enable) set ntp timezone PST -8
Timezone set to "PST", offset from UTC is -8 hours.
Console> (enable)
```

**Related Commands** [clear ntp timezone](#)  
[show ntp](#)

# set password

To change the login password on the CLI, use the **set password** command.

## **set password**

---

**Syntax Description** This command has no arguments or keywords.

---

**Defaults** The default is no password is configured.

---

**Command Types** Switch command.

---

**Command Modes** Privileged.

---

**Usage Guidelines** Passwords are case sensitive and may be from 0 to 19 characters in length, including spaces. The command prompts you for the old password. If the password you enter is valid, you are prompted to enter a new password and to verify the new password. A zero-length password is allowed by pressing **Return**.

---

**Examples** This example shows how to set an initial password:

```
Console> (enable) set password
Enter old password: <old_password>
Enter new password: <new_password>
Retype new password: <new_password>
Password changed.
Console> (enable)
```

# set pbf

To enable policy-based forwarding (PBF) and to set a MAC address for the PFC2, use the **set pbf** command.

```
set pbf [mac mac_address]
```

---

## Syntax Description

**mac mac\_address** (Optional) Specifies MAC address for the PFC2.

---



---

## Defaults

You can use the default MAC address, or you can specify a MAC address. See the “Usage Guidelines” section for more information.

---

## Command Types

Switch command.

---

## Command Modes

Privileged.

---

## Usage Guidelines

You must set a MAC address for the PFC2. We recommend that you use the default MAC address provided by the MAC PROM. When you specify your own MAC address using the **set pbf mac** command, if the MAC address is a duplicate of a MAC address already in use, packets might be dropped.

PBF is not supported with an operating (booted) MSFC2 in the Catalyst 6500 series switch that is being used for PBF. If an MSFC2 is present but not booted, you can configure PBF.

PBF may require some configuration on attached hosts. When a router is not present in the network, ARP table entries have to be statically added on each host participating in PBF. Refer to the “Configuring Policy-Based Forwarding” section of Chapter 16, “Configuring Access Control,” in the *Catalyst 6500 Series Switch Software Configuration Guide* for detailed information on configuring hosts.



### Note

PBF does not work with 802.1Q tunnel traffic. PBF is supported on Layer 3 IP unicast traffic, but it is not applicable to Layer 2 traffic. At the intermediate (PBF) switch, all 802.1Q tunnel traffic appears as Layer 2 traffic.

---



---

## Examples

This example shows how to set the default MAC address for the PFC2:

```
Console> (enable) set pbf
Console> (enable) Operation successful.
Console> (enable)
```

This example shows how to set a specific MAC address for the PFC2:

```
Console> (enable) set pbf mac 00-01-64-61-39-c2
Console> (enable) Operation successful.
Console> (enable)
```

**Related Commands**

[clear pbf](#)  
[show pbf](#)

# set pbf-map

To create security ACLs and to set adjacency information, use the **set pbf-map** command.

```
set pbf-map {ip_addr_1} {mac_addr_1} {vlan_1} {ip_addr_2} {mac_addr_2} {vlan_2}
```

Syntax Description		
<i>ip_addr_1</i>		IP address of host 1.
<i>mac_addr_1</i>		MAC address of host 1.
<i>vlan_1</i>		Number of the first VLAN.
<i>ip_addr_2</i>		IP address of host 2.
<i>mac_addr_2</i>		MAC address of host 2.
<i>vlan_2</i>		Number of the second VLAN.

**Defaults** This command has no default settings.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** The **set pbf-map** command does not change existing commands or NVRAM.

The **set pbf-map** command creates security ACLs and adjacency information based on your input and then automatically commits the ACLs. This command simplifies the configuration of policy-based forwarding.

An example of the simplified syntax is **set pbf-map 1.1.1 0-0-0-0-1 11 2.2.2 0-0-0-0-2 12**.

The above example is equivalent to all of the following PBF commands, which were released prior to 7.4:

```
set security acl adjacency PBF_MAP_ADJ_0 11 0-0-0-0-1
set security acl adjacency PBF_MAP_ADJ_1 12 0-0-0-0-2
commit security acl adjacency
set security acl ip PBF_MAP_ACL_11 redirect PBF_MAP_ADJ_1 ip host 1.1.1 host 2.2.2
set security acl ip PBF_MAP_ACL_12 redirect PBF_MAP_ADJ_0 ip host 2.2.2 host 1.1.1
```

If the **permit ip any any** ACE is missing, the following two entries are added:

```
set security acl ip PBF_MAP_ACL_11 permit ip any any
set security acl ip PBF_MAP_ACL_12 permit ip any any
commit security acl ip PBF_MAP_ACL_11
commit security acl ip PBF_MAP_ACL_12
set security acl map PBF_MAP_ACL_11 11
set security acl map PBF_MAP_ACL_12 12
```

Each entry in the ACL that is added by the **set pbf-map** command is inserted before the default **permit ip any any** ACE.

If you want to add entries other than redirect ACEs to the adjacency table, use the **set security acl ip PBF\_MAP\_ACL\_(VLAN\_ID)** command.

### Examples

This example shows how to specify a PBF\_MAP\_ACL:

```
Console> (enable) set pbf-map 1.1.1.1 0-0-0-0-1 11 2.2.2.2 0-0-0-0-2 22

Commit operation successful.
Commit operation successful.

ACL 'PBF_MAP_ACL_11' successfully committed.
Console> (enable)
ACL PBF_MAP_ACL_11 successfully mapped to VLAN 11.
Console> (enable)
ACL 'PBF_MAP_ACL_22' successfully committed.
Console> (enable)
ACL PBF_MAP_ACL_22 successfully mapped to VLAN 22.
Console> (enable) Operation successful.
Console> (enable)
```

### Related Commands

[clear pbf-map](#)  
[show pbf-map](#)

## set port arp-inspection

To set Address Recognition Protocol (ARP) inspection thresholds on a per-port basis, use the **set port arp-inspection** command.

**set port arp-inspection** *mod/port* **drop-threshold** *rate* **shutdown-threshold** *rate*

<b>Syntax Description</b>	<i>mod/port</i>	Number of the module and port on the module.
	<b>drop-threshold</b>	Indicates the drop threshold.
	<i>rate</i>	Number of packets per second; valid values are from 0 to 1000 pps.
	<b>shutdown-threshold</b>	Indicates the shutdown threshold.

**Defaults** Both threshold rates are 0 packets per second.

**Command Types** Switch command

**Command Modes** Privileged.

**Usage Guidelines** If the number of packets exceeds the drop-threshold rate, the excess packets are dropped. The excess packets are still counted toward the shutdown-threshold rate. If the number of packets exceeds the shutdown-threshold rate, the port is shut down.

When the threshold rates are both at 0 packets per second, per-port rate limiting is not on.

**Examples** This example shows how to set the drop-threshold to 500 and the shutdown-threshold to 1000 for port 2/1:

```
Console> (enable) set port arp-inspection 2/1 drop-threshold 500 shutdown-threshold 1000
Drop Threshold=500, Shutdown Threshold=1000 set on port 2/1.
Console> (enable)
```

**Examples** [set security acl arp-inspection](#)  
[show port arp-inspection](#)