



Checking Status and Connectivity

This chapter describes how to check switch port status and connectivity on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Command Reference* publication.

This chapter consists of these sections:

- [Checking Module Status, page 19-1](#)
- [Checking Port Status, page 19-2](#)
- [Checking the Port MAC Address, page 19-4](#)
- [Checking Port Capabilities, page 19-4](#)
- [Checking the 10-Gigabit Ethernet Link Status, page 19-5](#)
- [Checking Cable Status Using the Time Domain Reflectometer, page 19-6](#)
- [Using Telnet, page 19-7](#)
- [Using Secure Shell Encryption for Telnet Sessions, page 19-7](#)
- [Monitoring User Sessions, page 19-8](#)
- [Using Ping, page 19-9](#)
- [Using Layer 2 Traceroute, page 19-11](#)
- [Using IP Traceroute, page 19-13](#)
- [Using System Warnings on Port Counters, page 19-14](#)

Checking Module Status

Catalyst 6500 series switches are multimodule systems. You can see what modules are installed, as well as the MAC address ranges and version numbers for each module, using the **show module** *[mod]* command. Specify a particular module number to see detailed information on that module.

This example shows how to check module status. The output shows that there is one supervisor engine and four additional modules installed in the chassis.

```

Console> (enable) show module
Mod Slot Ports Module-Type           Model           Status
-----
1 1 2 1000BaseX Supervisor      WS-X6K-SUP1-2GE ok
2 2 24 100BaseFX MM Ethernet     WS-X6224-100FX-MT ok
3 3 8 1000BaseX Ethernet       WS-X6408-GBIC   ok
4 4 48 10/100BaseTX (Telco)     WS-X6248-TEL    ok
5 5 48 10/100BaseTX (RJ-45)     WS-X6248-RJ-45  ok

Mod Module-Name           Serial-Num
-----
1                          SAD03040546
2                          SAD03110020
3                          SAD03070194
4                          SAD03140787
5                          SAD03181291

Mod MAC-Address (es)      Hw      Fw      Sw
-----
1 00-50-f0-a8-26-b2 to 00-50-f0-a8-26-b3 1.4 5.1(1) 5.2(1) CSX
  00-50-f0-a8-26-b0 to 00-50-f0-a8-26-b1
  00-50-3e-8d-64-00 to 00-50-3e-8d-67-ff
2 00-50-54-6c-e9-a8 to 00-50-54-6c-e9-bf 1.3 4.2(0.24)V 5.2(1) CSX
3 00-50-54-6c-93-6c to 00-50-54-6c-93-73 1.4 4.2(0.24)V 5.2(1) CSX
4 00-50-54-bf-59-64 to 00-50-54-bf-59-93 0.103 4.2(0.24)V 5.2(1) CSX
5 00-50-f0-ac-30-54 to 00-50-f0-ac-30-83 1.0 4.2(0.24)V 5.2(1) CSX

Mod Sub-Type              Sub-Model      Sub-Serial  Sub-Hw
-----
1 L2 Switching Engine I   WS-F6020      SAD03040312 1.0
Console> (enable)

```

This example shows how to check module status on a specific module:

```

Console> (enable) show module 4
Mod Slot Ports Module-Type           Model           Status
-----
4 4 48 10/100BaseTX (Telco)     WS-X6248-TEL    ok

Mod Module-Name           Serial-Num
-----
4                          SAD03140787

Mod MAC-Address (es)      Hw      Fw      Sw
-----
4 00-50-54-bf-59-64 to 00-50-54-bf-59-93 0.103 4.2(0.24)V 5.2(1) CSX
Console> (enable)

```

Checking Port Status

You can see summary or detailed information on the switch ports using the **show port** [*mod[/port]*] command. To see summary information on all of the ports on the switch, enter the **show port** command with no arguments. Specify a particular module number to see information on the ports on that module only. Enter both the module number and the port number to see detailed information about the specified port.

To apply configuration commands to a particular port, you must specify the appropriate logical module. For more information, see the [“Checking Module Status” section on page 19-1](#).

This example shows how to see information on the ports on a specific module only:

```

Console> (enable) show port 1
Port Name                Status      Vlan      Duplex Speed Type
-----
 1/1                    connected  1         full   1000 1000BaseSX
 1/2                    notconnect 1         full   1000 1000BaseSX

Port Security Secure-Src-Addr Last-Src-Addr Shutdown Trap IfIndex
-----
 1/1 disabled
 1/2 disabled                               No     disabled 3

Port Broadcast-Limit Broadcast-Drop
-----
 1/1 - 0
 1/2 - 0

Port Send FlowControl Receive FlowControl RxPause TxPause
      admin oper      admin oper
-----
 1/1 desired off      off off      0 0
 1/2 desired off      off off      0 0

Port Status Channel Admin Ch Neighbor Neighbor
      Mode      Group Id Device      Port
-----
 1/1 connected auto 65 0
 1/2 notconnect auto 65 0

Port Align-Err FCS-Err Xmit-Err Rcv-Err UnderSize
-----
 1/1 0 0 0 0 0
 1/2 0 0 0 0 0

Port Single-Col Multi-Coll Late-Coll Excess-Col Carri-Sen Runts Giants
-----
 1/1 0 0 0 0 0 0 0
 1/2 0 0 0 0 0 0 0

Last-Time-Cleared
-----
Tue Jun 8 1999, 10:01:35
Console> (enable)

```

This example shows how to see information on an individual port:

```

Console> (enable) show port 1/1
Port Name                Status      Vlan      Duplex Speed Type
-----
 1/1                    connected  1         full   1000 1000BaseSX

Port Security Secure-Src-Addr Last-Src-Addr Shutdown Trap IfIndex
-----
 1/1 disabled                               No     disabled 3

Port Broadcast-Limit Broadcast-Drop
-----
 1/1 - 0

Port Send FlowControl Receive FlowControl RxPause TxPause
      admin oper      admin oper
-----
 1/1 desired off      off off      0 0

```

```

Port    Status      Channel      Admin Ch      Neighbor      Neighbor
-----  -----  -
1/1    connected  auto        65    0            Device        Port

Port    Align-Err  FCS-Err      Xmit-Err      Rcv-Err      UnderSize
-----  -----  -
1/1          0          0            0            0            0

Port    Single-Col  Multi-Coll  Late-Coll      Excess-Col  Carri-Sen  Runts      Giants
-----  -----  -
1/1          0          0            0            0            0            0            0

Last-Time-Cleared
-----
Tue Jun 8 1999, 10:01:35
Console> (enable)

```

Checking the Port MAC Address

In addition to displaying the MAC address range for a module using the **show module** command, you can display the MAC address of a specific port in the switch using the **show port mac-address** *[mod[/port]]* command.

This example shows you how to display the MAC address of a specific port:

```

Console> show port mac-address 4/1
Port    Mac address
-----
4/1    00-50-54-bf-59-64

```

This example shows you how to display the MAC addresses of all ports on a module:

```

Console> show port mac-address 4
Port    Mac address
-----
4/1    00-50-54-bf-59-64
4/2    00-50-54-bf-59-65
4/3    00-50-54-bf-59-66
4/4    00-50-54-bf-59-67
...
4/47   00-50-54-bf-59-92
4/48   00-50-54-bf-59-93

```

Checking Port Capabilities

You can display the capabilities of any port in a switch using the **show port capabilities** *[[mod]/[port]]* command.

This example shows you how to display the port capabilities for switch ports:

```

Console> (enable) show port capabilities 1/1
Model                               WS-X6K-SUP1A-2GE
Port                                 1/1
Type                                 No Connector
Speed                                1000
Duplex                                full
Trunk encap type                     802.1Q, ISL
Trunk mode                            on,off,desirable,auto,nonegotiate
Channel                               yes
Broadcast suppression                percentage(0-100)
Flow control                          receive-(off,on,desired),send-(off,on,desired)
Security                              yes
Membership                            static,dynamic
Fast start                            yes
QOS scheduling                        rx-(1p1q4t),tx-(1p2q2t)
CoS rewrite                           yes
ToS rewrite                           DSCP
UDLD                                  yes
Inline power                          no
AuxiliaryVlan                        no
SPAN                                  source,destination
COPS port group                      1/1-2
Console> (enable)

```

Checking the 10-Gigabit Ethernet Link Status

Cable diagnostics allow you to activate the pseudorandom binary sequence (PRBS) test on 10-Gigabit Ethernet links.



Note

The PRBS test is currently available only on the 1-port 10GBASE-E serial 10-Gigabit Ethernet module (WS-X6502-10GE).

To run the PRBS test properly between two devices, you must start it on both ends of the cable. If the cable is looped back, a single end can generate the test sequence (on the Tx), verify the test sequence, and count the errors (at the Rx).

Before the PRBS test starts, the port is automatically put in errdisable state. The errdisable timeout is disabled for the port so that the port is not automatically reenabled after the timeout interval concludes. The errdisable timeout is automatically reenabled on the port after the PRBS test finishes.

When the PRBS test is running, the system does not permit you to enter the **set port enable** and **set port disable** commands.

The PRBS error counter measures the reliability of the cable. The error counter range is from 0–255. A value of 0 signifies a perfect link connection; a value of 255 signifies that the port is faulty, not connected, or that there is no communication through the link. If the counter does not remain at 0 for a predetermined length of time, the link is faulty. For example, for a baud error rate (BER) of 10^{-12} , the counter should remain at 0 for 100 seconds.

Each time that you access the PRBS counter by entering the **show port prbs** command, the PRBS error counter value is reset to 0, and the counter begins to accumulate errors again.



Note

The PRBS counter is a “read and clear” register. The first reading in a sequence is usually unreliable and serves primarily to purge the counter; successive readings are accurate.

To start or stop the PRBS test, perform this task in privileged mode:

	Task	Command
Step 1	Start or stop the PRBS test.	test cable-diagnostics prbs {start stop} mod/port
Step 2	Show the PRBS test counter information.	show port prbs

This example shows how to start the PRBS test on port 1 on module 5:

```
Console> (enable) test cable-diagnostics prbs start 5/1
PRBS cable-diagnostic test started on port 5/1.
Console> (enable)
```

This example shows how to stop the PRBS test on port 1 on module 5:

```
Console> (enable) test cable-diagnostics prbs stop 5/1
PRBS cable-diagnostic test stopped on port 5/1.
Console> (enable)
```

This example shows the message that displays when the PRBS test is not supported on a module:

```
Console> (enable) test cable-diagnostics prbs start 6/1
Feature not supported on module 6.
Console> (enable)
```

This example shows how to display PRBS counter values and the ports that are running the PRBS test:

```
Console> (enable) show port prbs

Max error counters = 255
Port    PRBS state    PRBS error counters
-----
6/1     start         30
7/1     stop          -
Console> (enable)
```

Checking Cable Status Using the Time Domain Reflectometer

You can check the status of copper cables using the time domain reflectometer (TDR) on the 48-port 10/100/1000 BASE-T modules for the Catalyst 6500 series switch (WS-X6148-GE-TX and WS-X6548-GE-TX). The TDR detects a cable fault by sending a signal through the cable and reading the signal that is reflected back to it. All or part of the signal can be reflected back by any number of cable defects or by the end of the cable itself.

Use TDR to determine if the cabling is at fault if you cannot establish a link. This is especially important when replacing an existing switch, upgrading to Gigabit Ethernet, or installing new cable plants.

To start or stop the TDR test, perform this task in privileged mode:

	Task	Command
Step 1	Start the TDR test.	test cable-diagnostics tdr mod/port
Step 2	Show the TDR test counter information.	show port tdr mod/port

This example shows how to start the TDR test on port 1 on module 2:

```

Console> (enable) test cable-diagnostics tdr 2/1
TDR test started on port 2/1. Use show port tdr <m/p> to see the results
Console> (enable)

```

This example shows how to display TDR test results for a port:

```

Console> (enable) show port tdr 2/1
TDR test last run on Mon, March 10 2003 at 1:35:00 pm
Port  Speed  Local pair  Pair length          Remote pair  Pair status
-----
2/1   1000    Pair A     12 +/- 3 meters     Pair A      Terminated
      Pair B     12 +/- 3 meters     Pair B      Terminated
      Pair C     12 +/- 3 meters     Pair C      Terminated
      Pair D     12 +/- 3 meters     Pair D      Terminated

```

Using Telnet

You can access the switch command-line interface (CLI) using Telnet. In addition, you can use Telnet from the switch to access other devices in the network. Up to eight simultaneous Telnet sessions are possible.

To Telnet to another device on the network from the switch, perform this task in privileged mode:

Task	Command
Open a Telnet session with a remote host.	telnet <i>host</i> [<i>port</i>]

This example shows how to Telnet from the switch to a remote host:

```

Console> (enable) telnet labsparc
Trying 172.16.10.3...
Connected to labsparc.
Escape character is '^]'.

UNIX(r) System V Release 4.0 (labsparc)

login:

```

Using Secure Shell Encryption for Telnet Sessions



Note

To use the Secure Shell encryption feature commands, you must be running an encryption image. The **set crypto key rsa**, **clear crypto key rsa**, and **show crypto key** commands are used for encryption. See [Chapter 25, “Working with System Software Images”](#) for the software image naming conventions that are used for the encryption images.

Secure Shell encryption provides security for Telnet sessions to the switch. Secure Shell encryption is supported for remote logins to the switch only. Telnet sessions that are initiated from the switch cannot be encrypted. To use this feature, you must install the application on the client accessing the switch, and you must configure Secure Shell encryption on the switch.

The current implementation of Secure Shell encryption supports SSH version 1 and the DES and 3DES encryption methods. Secure shell encryption can be used with RADIUS and TACACS+ authentication. To configure authentication with Secure Shell encryption, use the **telnet** keyword in the **set authentication** commands.

**Note**

If you are using Kerberos to authenticate to the switch, you will not be able to use Secure Shell encryption.

To enable Secure Shell encryption on the switch, perform this task in privileged mode:

Task	Command
Create the RSA host key.	set crypto key rsa <i>nbits</i> [force]

This example shows how to create the RSA host key:

```
Console> (enable) set crypto key rsa 1024
Generating RSA keys.... [OK]
Console> (enable)
```

The *nbits* value specifies the RSA key size. The valid key size range is from 512–2048 bits. A key size with a larger number provides higher security but takes longer to generate.

You can enter the optional **force** keyword to regenerate the keys and suppress the warning prompt of overwriting existing keys.

Monitoring User Sessions

You can display the currently active user sessions on the switch using the **show users** command. The command output displays all active console port and Telnet sessions on the switch.

To display the active user sessions on the switch, perform this task in privileged mode:

Task	Command
Display the currently active user sessions on the switch.	show users [noalias]

This example shows the output of the **show users** command when local authentication is enabled for console and Telnet sessions (the asterisk [*] indicates the current session):

```
Console> (enable) show users
  Session  User                Location
  -----
  console
  telnet                sam-pc.bigcorp.com
  * telnet              jake-mac.bigcorp.com
Console> (enable)
```

This example shows the output of the **show users** command when TACACS+ authentication is enabled for console and Telnet sessions:

```
Console> (enable) show users
  Session  User                Location
  -----
  console  sam
  telnet   jake                jake-mac.bigcorp.com
  telnet   tim                tim-nt.bigcorp.com
  * telnet  suzy                suzy-pc.bigcorp.com
Console> (enable)
```

This example shows how to display information about user sessions using the **noalias** keyword to display the IP addresses of connected hosts:

```
Console> (enable) show users noalias
  Session  User                Location
  -----
  console
  telnet           10.10.10.12
  * telnet         10.10.20.46
Console> (enable)
```

To disconnect an active user session, perform this task in privileged mode:

Task	Command
Disconnect an active user session on the switch.	disconnect {console ip_addr}

This example shows how to disconnect an active console port session and an active Telnet session:

```
Console> (enable) show users
  Session  User                Location
  -----
  console  sam
  telnet   jake                jake-mac.bigcorp.com
  telnet   tim                tim-nt.bigcorp.com
  * telnet  suzy                suzy-pc.bigcorp.com
Console> (enable) disconnect console
Console session disconnected.
Console> (enable) disconnect tim-nt.bigcorp.com
Telnet session from tim-nt.bigcorp.com disconnected. (1)
Console> (enable) show users
  Session  User                Location
  -----
  telnet   jake                jake-mac.bigcorp.com
  * telnet  suzy                suzy-pc.bigcorp.com
Console> (enable)
```

Using Ping

These sections describe how to use IP ping:

- [Understanding How Ping Works, page 19-10](#)
- [Executing Ping, page 19-10](#)

Understanding How Ping Works

You can use IP ping to test connectivity to remote hosts. If you attempt to ping a host in a different IP subnetwork, you must define a static route to the network or configure a router to route between those subnets.

The **ping** command is configurable from normal EXEC mode and privileged EXEC mode. In normal EXEC mode, the **ping** command supports the **-s** parameter, which allows you to specify the packet size and packet count. In privileged EXEC mode, the **ping** command lets you specify the packet size, packet count, and the wait time.

Table 19-1 shows the default values that apply to the **ping-s** command.

Table 19-1 Ping Default Values

Description	Ping	Ping-s
Number of Packets	5	0=continuous ping
Packet Size	56	56
Wait Time	2	2
Source Address	Host IP Address	N/A

To stop a ping in progress, press **Ctrl-C**.

Ping returns one of the following responses:

- Normal response—The normal response (*hostname* is alive) occurs in 1 to 10 seconds depending on network traffic.
- Destination does not respond—If the host does not respond, a no answer message is returned.
- Unknown host—If the host does not exist, an unknown host message is returned.
- Destination unreachable—If the default gateway cannot reach the specified network, a destination unreachable message is returned.
- Network or host unreachable—If there is no entry in the route table for the host or network, a network or host unreachable message is returned.

Executing Ping

To ping another device on the network from the switch, perform one of these tasks in normal or privileged mode:

Task	Command
Ping a remote host.	ping <i>host</i>
Ping a remote host using ping options.	ping -s <i>host</i> [<i>packet_size</i>] [<i>packet_count</i>]

This example shows how to ping a remote host from normal EXEC mode:

```
Console> ping labsparc
labsparc is alive
Console> ping 72.16.10.3
12.16.10.3 is alive
Console>
```

This example shows how to ping a remote host using the ping -s option:

```
Console> ping -s 12.20.5.3 800 10
PING 12.20.2.3: 800 data bytes
808 bytes from 12.20.2.3: icmp_seq=0. time=2 ms
808 bytes from 12.20.2.3: icmp_seq=1. time=3 ms
808 bytes from 12.20.2.3: icmp_seq=2. time=2 ms
808 bytes from 12.20.2.3: icmp_seq=3. time=2 ms
808 bytes from 12.20.2.3: icmp_seq=4. time=2 ms
808 bytes from 12.20.2.3: icmp_seq=5. time=2 ms
808 bytes from 12.20.2.3: icmp_seq=6. time=2 ms
808 bytes from 12.20.2.3: icmp_seq=7. time=2 ms
808 bytes from 12.20.2.3: icmp_seq=8. time=2 ms
808 bytes from 12.20.2.3: icmp_seq=9. time=3 ms

----17.20.2.3 PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 2/2/3
Console>
```

This example shows how to enter a ping command in privileged mode specifying the number of packets, the packet size, and the timeout period:

```
Console> (enable) ping
Target IP Address []: 12.20.5.19
Number of Packets [5]: 10
Datagram Size [56]: 100
Timeout in seconds [2]: 10
Source IP Address [12.20.2.18]: 12.20.2.18
!!!!!!!!!!!!

----12.20.2.19 PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 1/1/1
Console> (enable)
```

Using Layer 2 Traceroute

The Layer 2 Traceroute utility allows you to identify the physical path that a packet will take when going from a source to a destination. The Layer 2 Traceroute utility determines the path by looking at the forwarding engine tables of the switches in the path.

Information is displayed about all Catalyst 6500 series switches that are in the path from the source to the destination.

These sections describe how to use Layer 2 Traceroute:

- [Layer 2 Traceroute Usage Guidelines, page 19-12](#)
- [Identifying a Layer 2 Path, page 19-12](#)

Layer 2 Traceroute Usage Guidelines

Follow these guidelines for using the Layer 2 Traceroute utility:

- The Layer 2 Traceroute utility works for unicast traffic only.
- You must enable CDP on all of the Catalyst 5000 and 6500 series switches in the network. (See [Chapter 29, “Configuring CDP”](#) for information about enabling CDP.) If any devices in the path are transparent to CDP, **l2trace** will not be able to trace the Layer 2 path through those devices.
- You can use this utility from a switch that is not in the Layer 2 path between the source and the destination; however, all of the switches in the path, including the source and destination, must be reachable from the switch.
- All switches in the path must be reachable from each other.
- You can trace a Layer 2 path by specifying the source and destination IP addresses (or IP aliases) or the MAC addresses. If the source and destination belong to multiple VLANs and you specify MAC addresses, you can also specify a VLAN.
- The source and destination switches must belong in the same VLAN.
- The maximum number of hops an **l2trace** query will try is 10; this includes hops involved in source tracing.
- The Layer 2 Traceroute utility does not work with Token Ring VLANs, when multiple devices are attached to one port through hubs, or when multiple neighbors are on a port.

Identifying a Layer 2 Path

To identify a Layer 2 path, perform one of these tasks in privileged mode:

Task	Command
(Optional) Trace a Layer 2 path using MAC addresses.	l2trace {src-mac-addr} {dest-mac-addr} [vlan] [detail]
(Optional) Trace a Layer 2 path using IP addresses or IP aliases.	l2trace {src-ip-addr} {dest-ip-addr} [detail]

This example shows the source and destination MAC addresses specified, with no VLAN specified, and the detail option specified. For each Catalyst 5000 and 6500 series switch found in the path, the output shows the device type, device name, device IP address, in port name, in port speed, in port duplex mode, out port name, out port speed, and out port duplex mode.

```
Console> (enable) l2trace 00-01-22-33-44-55 10-22-33-44-55-66 detail
```

```
l2trace vlan number is 10.
```

```
00-01-22-33-44-55 found in C5500 named wiring-1 on port 4/1 10Mb half duplex
C5500:wiring-1:192.168.242.10:4/1 10Mb half duplex -> 5/2 100MB full duplex
C5000:backup-wiring-1:192.168.242.20:1/1 100Mb full duplex -> 3/1 100MB full duplex
C5000:backup-core-1:192.168.242.30:4/1 100 MB full duplex -> 1/1 100MB full duplex
C6000:core-1:192.168.242.40:1/1 100MB full duplex -> 2/1 10MB half duplex.
10-22-33-44-55-66 found in C6000 named core-1 on port 2/1 10MB half duplex.
```

Using IP Traceroute

The IP Traceroute utility allows you to identify the path that packets take through the network at Layer 3 on a hop-by-hop basis. The command output displays all network layer (Layer 3) devices, such as routers, that the traffic passes through on the way to the destination.

These sections describe how to use IP Traceroute:

- [Understanding How IP Traceroute Works, page 19-13](#)
- [Executing IP Traceroute, page 19-13](#)

Understanding How IP Traceroute Works

The **traceroute** command uses the Time To Live (TTL) field in the IP header to cause routers and servers to generate specific return messages. Traceroute starts by sending a User Datagram Protocol (UDP) datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it drops the datagram and sends back an Internet Control Message Protocol (ICMP) time-exceeded message to the sender. The traceroute facility determines the address of the first hop by examining the source address field of the ICMP time-exceeded message.

To identify the next hop, traceroute sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the time-exceeded message to the source. This process continues until the TTL is incremented to a value large enough for the datagram to reach the destination host (or until the maximum TTL is reached).

To determine when a datagram reaches its destination, traceroute sets the UDP destination port in the datagram to a very large value which the destination host is unlikely to be using. When a host receives a datagram with an unrecognized port number, it sends an ICMP port unreachable error to the source. This message indicates to the traceroute facility that it has reached the destination.

Switches can participate as the source or destination of the **traceroute** command but will not appear as a hop in the **traceroute** command output.

Executing IP Traceroute

To trace the path that packets take through the network, perform this task in privileged mode:

Task	Command
Execute IP traceroute to trace the Layer 3 path that packets take through the network.	traceroute [-n] [-w <i>wait_time</i>] [-i <i>initial_ttl</i>] [-m <i>max_ttl</i>] [-p <i>dest_port</i>] [-q <i>nqueries</i>] [-t <i>tos</i>] <i>host</i> [<i>data_size</i>]

This example shows how to use the **traceroute** command:

```
Console> (enable) traceroute 10.1.1.100
traceroute to 10.1.1.100 (10.1.1.100), 30 hops max, 40 byte packets
 1 10.1.1.1 (10.1.1.1)  1 ms  2 ms  1 ms
 2 10.1.1.100 (10.1.1.100)  2 ms  2 ms  2 ms
Console> (enable)
```

This example shows how to perform a **traceroute** with six queries to each hop with packets of 1400 bytes each:

```
Console> (enable) traceroute -q 6 10.1.1.100 1400
traceroute to 10.1.1.100 (10.1.1.100), 30 hops max, 1440 byte packets
 1 10.1.1.1 (10.1.1.1)  2 ms  2 ms  2 ms  1 ms  2 ms  2 ms
 2 10.1.1.100 (10.1.1.100)  2 ms  4 ms  3 ms  3 ms  3 ms  3 ms
Console> (enable)
```

Using System Warnings on Port Counters

You can monitor and troubleshoot the Catalyst 6500 series switches by polling selected error counters on all ports and logging the system error messages. Messages are logged for the system, hardware, and spanning tree ports for these conditions:

- Backplane traffic levels that exceed configurable thresholds
- Low remaining memory
- Detected memory corruption
- NVRAM logs
- Inband errors
- User Datagram Protocol (UDP) and TCP errors

Hardware error information is logged to provide information for debug port counters at 30-minute intervals. Messages are logged if the counter values increase.

Spanning tree error information is provided for the following:

- Ports that go from the blocking to the forwarding state
- Bridge protocol data unit (BPDU) skewing that exceeds a fixed threshold

These sections describe how to use the system warning feature on the Catalyst 6500 series switches:

- [Executing System Warnings on Port Counters, page 19-14](#)
- [Executing Hardware Level Warnings on Port Counters, page 19-17](#)
- [Executing Spanning Tree Warnings on Port Counters, page 19-18](#)

Executing System Warnings on Port Counters

These sections describe how to execute the system warnings on port counters:

- [Backplane Traffic, page 19-15](#)
- [Low Remaining Memory, page 19-16](#)
- [Detected Memory Corruption, page 19-16](#)
- [NVRAM Logs, page 19-16](#)
- [Inband Errors, page 19-17](#)
- [UDP Errors, page 19-17](#)

Backplane Traffic

You can configure backplane threshold detection by using a high threshold as a percentage. When backplane traffic goes over the specified threshold, compared with the previous traffic poll, a syslog message is generated. However, if you specify a 100-percent threshold (the default), no syslog message is generated.

For switches with three switching buses, you can configure a threshold and syslog throttling (to control the syslog event polling and message generation) for each switching bus instead of configuring the average traffic of all three buses. The throttle interval is 5 minutes.

This example shows how to set a threshold:

```

Console> (enable) set traffic monitor help
Usage: set traffic monitor <threshold>
      (threshold = 0..100 in percentage)
Console> (enable) set traffic monitor 60
Traffic monitoring threshold set to 60%.
Console> (enable) show traffic
Threshold: 60%

Backplane-Traffic Peak Peak-Time
-----
0%                0% Tue Apr 16 2002, 08:01:53

Fab Chan Input Output
-----
0      0%    0%
1      0%    0%
2      0%    0%
3      0%    0%
4      0%    0%
5      0%    0%
6      0%    0%
7      0%    0%
8      0%    0%
9      0%    0%
10     0%    0%
11     0%    0%
12     0%    0%
13     0%    0%
14     0%    0%
15     0%    0%
16     0%    0%
17     0%    0%
Console> (enable)

```

Some sample syslog messages are as follows:

```

2000 Jan 11 06:00:27 PST -07:00 %SYS-4-SYS_HITRFC: 62% traffic detected on switching bus
(A)
2000 Feb 21 12:00:27 PST -07:00 %SYS-4-SYS_HITRFC: 65% traffic detected on switching bus

```

Low Remaining Memory

When memory allocation of clusters and buffers on the Catalyst 6500 series switch goes above a high watermark of 90 percent, syslog messages are generated. These actions generate syslog messages:

- When cluster allocation usage goes above a high watermark of 90 percent, the throttle interval is 1 hour.
- When mbufs allocation usage goes above a high watermark of 90 percent, the throttle interval is 1 hour.
- When malloc allocation usage goes above a high watermark of 90 percent, the throttle interval is 1 hour.

A sample syslog message is as follows:

```
1999 Sep 9 00:00:00 PDT -07:00 %SYS-3-SYS_MEMLOW: Memory cluster usage exceeded 90%
1999 Sep 9 00:00:00 PDT -07:00 %SYS-3-SYS_MEMLOW: Mbuf usage exceeded 90%
1999 Sep 9 00:00:00 PDT -07:00 %SYS-3-SYS_MEMLOW: Malloc usage exceeded 90%
```

Detected Memory Corruption

By default, memory corruption that is detected by the Memory Management Module (MMU) is disabled. This example shows how to enable memory corruption detection:



Note

In the 7.x release train, memory corruption that is detected by the MMU is enabled by default in software release 7.6(12) and later releases.

```
Console> (enable) set errordetection memory
Usage: set errordetection memory <enable|disable>
Console> (enable) set errordetection memory enable
Memory error detection enabled.
Console> (enable) show errordetection
Memory: enabled
Inband: disabled
```

A sample syslog message is as follows:

```
1999 Nov 23 16:32:21 PDT -07:00 %SYS-3-SYS_MEMERR: Out of range while freeing address
0xabcddefab
```

NVRAM Logs

Syslog errors are generated for each configuration-related NVRAM log event. These events may indicate configuration or hardware errors or NVRAM configurations that are made without notification of users. The hardware errors NVRAM log is not syslogged. The NVRAM log time stamp is not included in the message.

A sample syslog message is as follows:

```
1999 Nov 23 16:37:21 PDT -07:00 %SYS-4-SYS_NVLOG: convert_post_SAC_CiscoMIB:Block 63
converted from version 0 to 1

1999 Nov 23 16:37:25 PDT -07:00 %SYS-4-SYS_NVLOG: StartupConfig:Auto config started
```

Inband Errors

Inband syslog messages are generated when transmit or receive errors are detected. By default, inband syslog messages are disabled. This example shows how to enable inband error detection:



Note

In the 7.x release train, inband syslog messages are enabled by default in software release 7.6(12) and later releases.

```
Console> (enable) set errordetection inband
Usage: set errordetection inband <enable|disable>
Conosle> (enable) set errordetection inband enable
Inband errordetection enabled.
```

When resource errors on the receive side reach a multiple of 500, this syslog error is generated:

```
2000 Jun 24 06:37:25 PDT -07:00 %SYS-3-INBAND_NORESOURCE: inband resource error warning (500)
2000 Jun 24 08:12:03 PDT -07:00 %SYS-3-INBAND_NORESOURCE: inband resource error warning (1000)
```

For each spurious interrupt, a message similar to the following is logged:

```
1999 Dec 25 18:22:08 PDT -07:00 %SYS-3-INBAND_SPRINTR: inband spurious interrupt occurred (2)
```

For each inband port transmit and receive failure, a message similar to the following is logged:



Note

The number in parentheses indicates the number of times that the inband port is reset instead of the number of transmit or receive fails.

```
1999 Dec 25 18:22:08 PDT -07:00 %SYS-3-INBAND_TXRXFAIL: inband driver stuck/reset (2)
```

UDP Errors

When you enter the **show netstat udp** command, each socket overflow generates a message similar to the following:

```
1999 Oct 31 23:59:59 PDT -07:00 %IP-3-UDP_SOCKOVFL: UDP socket overflow
```

When you enter the **show netstat udp/tcp** command, each bad UDP/TCP checksum generates a message similar to the following:

```
1999 Oct 31 23:59:59 PDT -07:00 %IP-3-UDP_BADCKSUM: UDP bad checksum
1999 Oct 31 23:59:59 PDT -07:00 %IP-3-TCP_BADCKSUM: TCP bad checksum
```

Executing Hardware Level Warnings on Port Counters

You can poll selected error counters of each switch port every 30 minutes. If the count goes up between two subsequent polls on the same port, the incidence is logged. Background polling is enabled or disabled by the **set errordetection portcounters** command. By default, polling is disabled.



Note

In the 7.x release train, polling is enabled by default in software release 7.6(12) and later releases.

Enter the **set errordetection portcounters** command as follows:

```
Console> (enable) set errordetection portcounters
Usage: set errordetection portcounters <enable|disable>
Console> (enable) set errordetection portcounters disable
Port Counters error detection disabled.
```

A sample syslog message is as follows:

```
1999 Jan 11 08:02:59 PDT -07:00 %SYS-3-PORT_ERR: Port 3/4 swBusResultEvent (12)
1999 Jan 11 09:03:03 PDT -07:00 %SYS-3-PORT_ERR: Port 3/4 swBusResultEvent (223)
1999 Jan 11 09:03:03 PDT -07:00 %SYS-4-PORT_WARN: Port 3/4 dmaTxFull (7) dmaRetry (33)
dmaLevel2Request(21)
```

Executing Spanning Tree Warnings on Port Counters

These sections describe how to execute spanning tree warnings on the port counters:

- [Blocking to Listening Transitions, page 19-18](#)
- [BPDU Skewing, page 19-18](#)
- [SNMP, page 19-18](#)

Blocking to Listening Transitions

A syslog message is generated whenever a port goes from blocking to listening. Spanning-tree state changes have existing syslog messages.

A sample syslog messages is as follows:

```
1999 Jan 03 00:02:59 PDT -07:00 %SPANTREE-5-PORTLISTEN: Port 3/4 state in vlan 1 changed
to listening
1999 Jan 03 00:02:59 PDT -07:00 %SPANTREE-5-TR_PORTLISTEN: Trcrf 101 in trbrf 102 state
changed to listening
```

BPDU Skewing

A syslog message is generated when the interval between two consecutive BPDUs that are received on a port exceeds the hello time interval by 10 seconds. The throttle interval is one message per port, per minute for all VLAN numbers.

A sample syslog messages is as follows:

```
1999 Jan 01 00:01:19 PDT -07:00 %SPANTREE-3-BPDUSKEW: Port 2/1 vlan 1 BPDU skewed
1999 Jan 01 00:05:19 PDT -07:00 %SPANTREE-3-BPDUSKEW: Port 2/5 vlan 1 BPDU skewed
1999 Jan 01 00:05:23 PDT -07:00 %SPANTREE-3-BPDUSKEW: Port 2/5 vlan 3 BPDU skewed
```

SNMP

A matching SNMP trap generation for each of the syslog warnings using the existing `clogMessageGenerated` trap is sent every time that any syslog message is generated.