



Checking Port Status and Connectivity

This chapter describes how to check switch port status and connectivity on the Catalyst 6000 family switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6000 Family Command Reference* publication.

This chapter consists of these sections:

- [Checking Module Status, page 19-1](#)
- [Checking Port Status, page 19-2](#)
- [Checking Port Capabilities, page 19-4](#)
- [Using Telnet, page 19-4](#)
- [Using Secure Shell Encryption for Telnet Sessions, page 19-5](#)
- [Monitoring User Sessions, page 19-6](#)
- [Using Ping, page 19-7](#)
- [Using Layer 2 Traceroute, page 19-9](#)
- [Using IP Traceroute, page 19-10](#)

Checking Module Status

Catalyst 6000 family 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
      Status 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
-----
1/1      connected   auto         65    0
-----
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 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)

```

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 used for the encryption images.

The Secure Shell encryption feature provides security for Telnet sessions to the switch. Secure Shell encryption is supported for remote logins to the switch only. Telnet sessions 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, the DES and 3DES encryption methods, and 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 the Secure Shell encryption feature.

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> [<i>force</i>]

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 512 to 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 19-7](#)
- [Executing Ping, page 19-8](#)

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 executive and privileged executive mode. In normal executive mode, the **ping** command supports the **-s** parameter, which allows you to specify the packet size and packet count. In privileged executive 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 executive 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 6000 family 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 19-9](#)
- [Identifying a Layer 2 Path, page 19-10](#)

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 6000 family 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, or 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 6000 family 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 19-10](#)
- [Executing IP Traceroute, page 19-11](#)

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)
```

