



Checking Port Status and Connectivity

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



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference—Catalyst 4000 Family, Catalyst 2948G, and Catalyst 2980G Switches*.

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 19-5](#)
- [Changing the Login Timer, page 19-6](#)
- [Using Secure Shell Encryption for Telnet Sessions, page 19-6](#)
- [Monitoring User Sessions, page 19-7](#)
- [Using Ping, page 19-8](#)
- [Using Layer 2 Traceroute, page 19-10](#)
- [Using IP Traceroute, page 19-11](#)

Checking Module Status

The Catalyst enterprise LAN switches are multimodule systems. You can see what modules are installed, as well as the MAC address ranges and version numbers for each module, by using the **show module** command. You can use the `[mod_num]` argument to specify a particular module number to see detailed information on that module.

The Catalyst 4912G, 2948G, and 2980G switches are fixed-configuration switches, but are logically modular. You must apply configuration commands to the appropriate module. For example, on a Catalyst 2948G series switch, the 24 Fast Ethernet ports belong logically to module 2.

This example shows how to check module status on a Catalyst 2948G switch:

```

Console> (enable) show module
Mod Slot Ports Module-Type           Model           Status
-----
1  1    0      Switching Supervisor   WS-X2948       ok
2  1    50      10/100/1000 Ethernet   WS-X2948G      ok

Mod Module-Name      Serial-Num
-----
1  Supervisor         JAB023807H1
2  Switch Ports       JAB023807H1

Mod MAC-Address(es)           Hw   Fw   Sw
-----
1  00-50-73-12-09-00 to 00-50-73-12-0c-ff 1.0   4.4(1)  5.1(1)
2  00-50-73-12-0c-9e to 00-50-73-12-0c-fd 1.0
Console> (enable)

```

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

```

Console> (enable) show module 3
Mod Slot Ports Module-Type           Model           Sub Status
-----
3  3    6      1000BaseX Ethernet     WS-X4306       no ok

Mod Module-Name      Serial-Num
-----
3                      JAB024000YY

Mod MAC-Address(es)           Hw   Fw   Sw
-----
3  00-10-7b-f6-b2-1a to 00-10-7b-f6-b2-1f 0.2
Console> (enable)

```

Checking Port Status

You can see summary or detailed information on the switch ports using the **show 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.

The Catalyst 4912G, 2948G, and 2980G switches are fixed-configuration switches but are logically modular. 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 about the ports on a specific module only:

```

Console> (enable) show port 3
Port Name              Status      Vlan      Level Duplex Speed Type
-----
3/1                    connected  10        normal full  1000 1000BaseSX
3/2                    connected  10        normal full  1000 1000BaseSX
3/3                    connected  20        normal full  1000 1000BaseSX
3/4                    connected  40        normal full  1000 1000BaseSX
3/5                    notconnect 1         normal full  1000 No GBIC
3/6                    notconnect 1         normal full  1000 No GBIC

Port Security Secure-Src-Addr  Last-Src-Addr  Shutdown Trap  IfIndex
-----
3/1 disabled
3/2 disabled

```

```

3/3 disabled No disabled 17
3/4 disabled No disabled 18
3/5 disabled No disabled 19
3/6 disabled No disabled 20

```

Port	Send FlowControl admin oper	Receive FlowControl admin oper	RxPause	TxPause	Unsupported opcodes
3/1	desired on	desired on	0	0	0
3/2	desired on	desired on	0	0	0
3/3	desired on	desired on	0	0	0
3/4	desired on	desired on	0	0	0
3/5	desired off	off off	0	0	0
3/6	desired off	off off	0	0	0

Port	Status	Channel mode	Channel status	Neighbor device	Neighbor port
3/1	connected	off	not channel		
3/2	connected	off	not channel		
3/3	connected	off	not channel		
3/4	connected	off	not channel		
3/5	notconnect	off	not channel		
3/6	notconnect	off	not channel		

Port	Align-Err	FCS-Err	Xmit-Err	Rcv-Err	UnderSize
3/1	-	0	0	0	0
3/2	-	0	0	0	0
3/3	-	0	0	0	0
3/4	-	0	0	0	0
3/5	-	0	0	0	0
3/6	-	0	0	0	0

Port	Single-Col	Multi-Coll	Late-Coll	Excess-Col	Carri-Sen	Runts	Giants
3/1	0	0	0	0	0	0	0
3/2	0	0	0	0	0	0	0
3/3	0	0	0	0	0	0	0
3/4	0	0	0	0	0	0	0
3/5	0	0	0	0	0	0	0
3/6	0	0	0	0	0	0	0

Last-Time-Cleared

```

-----
Fri Apr 30 1999, 18:54:17
Console> (enable)

```

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

```

Console> (enable) show port 2/1
Port Name Status Vlan Level Duplex Speed Type
-----
2/1 inactive 100 normal auto auto 10/100BaseTX

Port AuxiliaryVlan AuxVlan-Status InlinePowered PowerAllocated
Admin Oper Detected mWatt mA @51V
-----
2/1 none none - - - -

Port Security Violation Shutdown-Time Age-Time Max-Addr Trap IfIndex
-----
2/1 disabled shutdown 0 0 1 disabled 15

```

```

Port  Num-Addr  Secure-Src-Addr  Age-Left  Last-Src-Addr  Shutdown/Time-Left
-----
 2/1          0                -          -            -              -

Port  Status      Channel          Admin Ch
-----
 2/1  inactive   auto silent          1    0

Port  Align-Err  FCS-Err  Xmit-Err  Rcv-Err  UnderSize
-----
 2/1          -         0        998      1012     0

Port  Single-Col  Multi-Coll  Late-Coll  Excess-Col  Carri-Sen  Runts  Giants
-----
 2/1          0          0          0          0          0       1012   0

Last-Time-Cleared
-----
Mon Jun 11 2001, 07:26:48
Console> (enable)

```

Checking Port Capabilities

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

This example shows you how to display the port capabilities for ports on module 2:

```

Console> (enable) show port capabilities 2
Model                WS-X4148
Port                 2/1
Type                 10/100BaseTX
Speed                auto,10,100
Duplex                half,full
Trunk encap type     802.1Q
Trunk mode            on,off,desirable,auto,nonegotiate
Channel              2/1-48
Flow control         no
Security              yes
Membership            static,dynamic
Fast start            yes
QOS scheduling        rx-(none),tx-(2q1t)
CoS rewrite           no
ToS rewrite           no
Rewrite               no
UDLD                  yes
Inline power          no
AuxiliaryVlan        1..1000,untagged,none
SPAN                  source,destination

-----
Model                WS-X4148
Port                 2/2
Type                 10/100BaseTX
Speed                auto,10,100
Duplex                half,full
Trunk encap type     802.1Q
Trunk mode            on,off,desirable,auto,nonegotiate
Channel              2/1-48
Flow control         no

```

```

Security                yes
Membership              static,dynamic
Fast start              yes
QoS scheduling          rx- (none) ,tx- (2q1t)
CoS rewrite             no
ToS rewrite             no
Rewrite                 no
UDLD                    yes
Inline power            no
AuxiliaryVlan           1..1000,untagged,none
SPAN                    source,destination

```

```

.
.
.

```

This example shows you how to display the port capabilities for port 5 on module 3:

```

Console> (enable) show port capabilities 3/5
Model                WS-X4148
Port                 3/5
Type                 10/100BaseTX
Speed                auto,10,100
Duplex               half,full
Trunk encap type     802.1Q
Trunk mode           on,off,desirable,auto,nonegotiate
Channel              3/1-48
Flow control         no
Security             yes
Membership           static,dynamic
Fast start           yes
QoS scheduling       rx- (none) ,tx- (2q1t)
CoS rewrite          no
ToS rewrite          no
Rewrite              no
UDLD                 yes
Inline power         no
AuxiliaryVlan        1..1000,untagged,none
SPAN                 source,destination

```

```

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.

Before you can open a Telnet session to the switch, you must first set the IP address (and in some cases the default gateway) for the switch. For information about setting the IP address and default gateway, see [Chapter 3, “Configuring the Switch IP Address and Default Gateway.”](#)

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

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

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

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

UNIX(r) System V Release 4.0 (labsparc)

login:
```

Changing the Login Timer

The login timer is the number of minutes after which an idle session is disconnected.

To change the logout timer value, perform this task in privileged mode:

Task	Command
Change the logout timer value (a timeout value of 0 prevents idle sessions from being disconnected automatically).	set logout <i>timeout</i>

This example shows how to set the logout timer value to 10 minutes:

```
Console> (enable) set logout 10
Sessions will be automatically logged out after 10 minutes of idle time.
Console> (enable)
```

This example shows how to set the logout timer value to 0, preventing idle sessions from being disconnected automatically:

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

Using Secure Shell Encryption for Telnet Sessions



Note

To use the secure shell encryption (SSH) feature commands, you must be running an encryption image. Encryption commands are **set crypto key rsa**, **clear crypto key rsa**, and **show crypto key**. See [Chapter 29, “Working with System Software Images,”](#) for the software image naming conventions used for the encryption images.

The SSH feature provides security for Telnet sessions to the switch. SSH 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 SSH the switch.

The current implementation of SSH supports version 1, both the DES and 3DES encryption methods, and can be used with RADIUS and TACACS+ authentication. To support authentication for Telnet 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 SSH 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.

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 19-8](#)
- [Executing Ping, page 19-9](#)

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 allows you to specify the packet size, packet count, and the wait time.

These default values apply to the **ping-s** command:

Table 19-1 Ping Default Values

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

Ping will return 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.

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

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 host
Ping a remote host using ping options.	ping -s host [packet_size] [packet_count]

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. This utility determines the path by looking at the forwarding engine tables of the switches in the path.

Information is displayed about all Catalyst 4000, 5000, and 6000 family switches that are in the path from the source to the destination.

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 4000, 5000 and 6000 family switches in the network. (See [Chapter 20, “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 to 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 via 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
Trace a Layer 2 path using MAC addresses.	l2trace {src-mac-addr} {dest-mac-addr} [vlan] [detail]
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 but with the detail option specified. For each Catalyst 4000, 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 C4000 named wiring-1 on port 4/1 10Mb half duplex
C4000:wiring-1:192.168.242.10:4/1 10Mb half duplex -> 5/2 100MB full duplex
C4000: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

These sections describe how to use IP traceroute:

- [Understanding How IP Traceroute Works, page 19-11](#)
- [Executing IP Traceroute, page 19-12](#)

Understanding How IP Traceroute Works

You can use IP traceroute to identify the path that packets take through the network 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.

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

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 that 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.

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 path 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 the basic usage of 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)
```