# Recover Errdisable Port State on Cisco IOS Platforms

## Contents


**Introduction**

**Prerequisites**

- Requirements
- Components Used

**Background Information**

**Errdisable**

- Function of Errdisable
- Causes of Errdisable
- Determine if Ports are in the Errdisabled State
- Determine the Reason for the Errdisabled State (Console Messages, Syslog, and the Show Errdisable Recovery Command)
- Recover a Port from Err-disabled State
  - Correct the Root Problem

**Verify**

**Troubleshoot**

**Related Information**


## Introduction

This document describes the errdisabled state, how to recover from it, and provides examples of errdisable recovery.

---

## Prerequisites

### Requirements

There are no specific requirements for this document.

### Components Used

The outputs in this document were taken from Cisco Catalyst 4500/6500 Series Switches. The switches were running Cisco IOS® Software and had Ethernet ports that are capable of EtherChannel and PortFast.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

---

# Background Information

This document uses the feature **errdisable** and **err-disabled** for the interface state. It is common to seek technical support when noticing one or more switch ports have become errdisable, which means the ports have a status of err-disabled. The goal of this document is to help understand why the errdisable occurred and how to restore the ports to normal operation.

---



**Note**: The port status of err-disabled displays in the output of the **show interfaces status** command.

---

The errdisable feature is supported on Catalyst switches that run Cisco IOS® and Cisco IOS XE; the commands and examples in this document focus on the Cisco IOS® Software platforms identified in Components Used.

The commands used to implement and verify errdisable can vary between software platforms. This document specifically focuses on errdisable for switches that run Cisco IOS Software.

---

# Errdisable

## Function of Errdisable

If the configuration shows a port to be enabled, but software on the switch detects an error situation on the port, the software shuts down that port. In other words, the port is automatically disabled by the switch operating system software because of an error condition that is encountered on the port.

When a port is err-disabled, it is effectively shut down and no traffic is sent or received on that port. The port LED is set to the color orange, and when you run the **show interfaces** command, the port status shows err-disabled. This is an example of what an error-disabled port looks like from the Command Line Interface (CLI) of the switch:

<#root>

```
cat6k#
```

**show interfaces gigabitethernet 4/1 status**

```
Port    Name       Status       Vlan       Duplex  Speed Type
Gi4/1              err-disabled 100          full   1000 1000BaseSX
```

---

Or, if the interface has been disabled because of an error condition, you can see messages that are similar to these in both the console and the syslog:

```
%SPANTREE-SP-2-BLOCK_BPDUGUARD: Received BPDU on port GigabitEthernet4/1 with BPDU Guard enabled. Disab

%PM-SP-4-ERR_DISABLE: bpduguard error detected on Gi4/1, putting Gi4/1 in err-disable state
```

---

This example message displays when a host port receives the bridge protocol data unit (BPDU). The actual message depends on the reason for the error condition.

The errdisable feature serves two purposes:

- It lets the administrator know when and where there is a port problem.
- It eliminates the possibility this port can cause other ports on the module (or the entire module) to fail.
    - Such a failure can occur when a bad port monopolizes buffers or port error messages monopolize interprocess communications on the card, which can cause serious network issues. The errdisable feature helps prevent these situations.

---

## Causes of Errdisable

This feature was first implemented to handle special collision situations if the switch detected excessive or late collisions on a port. Excessive collisions occur when a frame is dropped because the switch encounters 16 collisions in a row. Late collisions occur because every device on the wire did not recognize the wire was in use. Possible causes of these types of errors include:

- A cable that is out of specification (either too long, the wrong type, or defective).
- A bad network interface card (NIC) card (with physical problems or driver problems).
- A port duplex misconfiguration.
    - A port duplex misconfiguration is a common cause of errors because of failures to negotiate the speed and duplex properly between two directly connected devices (such as, a NIC that connects to a switch).
    - Only half-duplex connections can ever have collisions in a LAN. Because of the carrier sense multiple access (CSMA) nature of Ethernet, collisions are normal for half duplex, as long as the

collisions do not exceed a small percentage of traffic.

---

There are various reasons for the interface to go into errdisable. These reasons can include:

- Duplex mismatch
- Port channel misconfiguration(s)
- BPDU guard violation(s)
- UniDirectional Link Detection (UDLD) condition(s)
- Late-collision detection(s)
- Link-flap detection(s)
- Security violation(s)
- Port Aggregation Protocol (PAgP) flap
- Layer 2 Tunneling Protocol (L2TP) guard
- DHCP snooping rate-limit
- Incorrect GBIC / Small Form-Factor Pluggable (SFP) module or cable
- Address Resolution Protocol (ARP) inspection
- Inline power

---

**Note**: Errdisable detection defaults depend on the platform and software release; check the configured detection status with **show errdisable detect**. In order to disable detection for a supported cause, specify that cause with **no errdisable detect cause <cause>** only after you assess the effect of removing the protective response.

---

## Determine if Ports are in the Errdisabled State

You can determine if your port has been errdisabled by running the **show interfaces <interface> status** command.

This is an example of an active port:

<#root>

```
cat6k#
```

**show interfaces gigabitethernet 4/1 status**

```
Port    Name               Status       Vlan       Duplex  Speed Type
Gi4/1                      connected    100          full   1000 1000BaseSX
```

---

In this next example, the same port is in the err-disabled state:

```
<#root>

cat6k#

show interfaces gigabitethernet 4/1 status


Port    Name               Status       Vlan       Duplex  Speed Type

Gi4/1                      err-disabled 100          full   1000 1000BaseSX
```

---

**Note**: When a port is err-disabled, the LED on the front panel that is associated with the port is set to the color orange.

---

## Determine the Reason for the Errdisabled State (Console Messages, Syslog, and the Show Errdisable Recovery Command)

When the switch puts a port in the error-disabled state, the switch sends a message to the console that describes why it disabled the port. The example in this section provides two sample messages that show the reason for port disablement:

- One disablement is because of the PortFast BPDU guard feature.
- The other disablement is because of an EtherChannel configuration problem.

---

**Note**: You can also see these messages in the syslog if you run the **show logging** command.

---

These are some sample messages:

```
%SPANTREE-SP-2-BLOCK_BPDUGUARD: Received BPDU on port GigabitEthernet4/1 with BPDU Guard enabled. Disab

%PM-SP-4-ERR_DISABLE: bpduguard error detected on Gi4/1, putting Gi4/1 in err-disable state

%SPANTREE-2-CHNMISCFG: STP loop - channel 11/1-2 is disabled in vlan 1
```

---

If you have enabled errdisable recovery, you can run the **show errdisable recovery** command to view the cause reported for interfaces listed for automatic recovery:

<#root>

```
cat6k#

show errdisable recovery

ErrDisable Reason    Timer Status
-----------------    --------------
udld                 Enabled
bpduguard            Enabled
security-violatio    Enabled
channel-misconfig    Enabled
pagp-flap            Enabled
dtp-flap             Enabled
link-flap            Enabled
l2ptguard            Enabled
psecure-violation    Enabled
gbic-invalid         Enabled
dhcp-rate-limit      Enabled
mac-limit            Enabled
unicast-flood        Enabled
arp-inspection       Enabled

Timer interval: 300 seconds

Interfaces that can be enabled at the next timeout:

Interface      Errdisable reason      Time left(sec)
---------    ---------------------    --------------
  Fa2/4                bpduguard          273
```

---

## Recover a Port from Err-disabled State

This section provides examples of how you can encounter an err-disabled port and how to fix it, and a few additional reasons that a port can become err-disabled. To recover a port from the err-disabled state, first identify and correct the root problem, then reenable the port. If you reenable the port before you fix the root problem, the ports become err-disabled again.

---

### Correct the Root Problem

After you discover why the ports were disabled, fix the root problem. The fix depends on what triggered the problem. There are numerous things that can trigger the shutdown. This section discusses some of the most noticeable and common causes:

- EtherChannel Misconfiguration:

- In order for EtherChannel to work, the ports involved must have consistent configurations. The ports must have the same VLAN, trunk mode, speed, duplex, and so on. Most of the configuration differences within a switch are caught and reported when you create the channel. If one switch is configured for EtherChannel and the other switch is not configured for EtherChannel, the spanning tree process can shut down the channeled ports on the side configured for EtherChannel.

- The on mode of EtherChannel does not send PAgP packets to negotiate with the other side before channeling; it assumes the other side is channeling. In addition, this example does not turn on EtherChannel for the other switch, it leaves these ports as individual, unchanneled ports. If you leave the other switch in this state for a minute or so, Spanning Tree Protocol (STP) on the switch where the EtherChannel is turned on thinks there is a loop. This puts the channeling ports in the errdisabled state.

---

In this example, a loop was detected and the ports were disabled. The output of the **show etherchannel summary** command shows the `Number of channel-groups in use` is `0`. When you look at one of the ports that are involved, you can see the status is `err-disabled`:

```
<#root>

%SPANTREE-2-CHNL_MISCFG: Detected loop due to etherchannel misconfiguration of Gi4/1

cat6k#

show etherchannel summary


Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        u - unsuitable for bundling
Number of channel-groups in use: 0
Number of aggregators:           0

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
```

---

The EtherChannel was torn down because the ports were placed in errdisable on this switch.

```
<#root>

cat6k#

show interfaces gigabitethernet 4/1 status


Port    Name               Status       Vlan       Duplex  Speed Type
```

```
Gi4/1                    err-disabled 100          full   1000 1000BaseSX
```

---

To determine what the problem was, look at the error message. The message indicates the EtherChannel encountered a spanning tree loop. As this section explains, this problem can occur when one device (the switch, in this case) has EtherChannel turned on manually with use of the on mode (as opposed to desirable). The other connected device (the other switch, in this case), does not have EtherChannel turned on at all. To resolve this, when both switches support PAgP, to set the channel mode to desirable on both sides of the connection, and then reenable the ports. Then each side forms a channel only if both sides agree to channel. If they do not form a channel, verify the resulting status of each member port before allowing traffic through the connection.

<#root>

```
cat6k(config)#
```

**interface gigabitethernet 4/1**

```
cat6k(config-if)#
```

**channel-group 3 mode desirable non-silent**

- Duplex Mismatch:
  - Duplex mismatches are common because of failures to auto-negotiate speed and duplex properly. Unlike a half duplex device, which must wait until there are no other devices that transmit on the same LAN segment, a full-duplex device transmits whenever the device has something to send, regardless of other devices. If this transmission occurs while the half-duplex device also transmits, the half-duplex device considers this either a collision (during the slot time) or a late collision (after the slot time). Because the full-duplex side never expects collisions, this side never realizes it must retransmit the dropped packet. A low percentage rate of collisions is normal with a half duplex, but is not normal with a full duplex.
  - A switch port that receives many late collisions usually indicates a duplex mismatch problem. Ensure both ends use compatible speed and duplex settings, including compatible auto-negotiation settings. The **show interfaces <interface>** command tells you the speed and duplex for Catalyst switch ports. Later versions of Cisco Discovery Protocol (CDP) can warn you on duplex mismatches before the port is put in the error-disabled state.

---

In addition, there are settings on a NIC, such as auto-polarity features that can cause the problem. Check the NIC documentation and disable a setting only if you identify it as the cause of the link problem. If you have multiple NICs from a vendor and the NICs all appear to have the same problem, check the manufacturer website for the release notes and ensure you have the latest drivers.

Other causes of late collisions include:

- A bad NIC (with physical problems, not just configuration problems)
- A bad cable

- A cable segment that is too long
- BPDU port guard

A port that uses PortFast must only connect to an end station (such as a workstation or server) and not to devices that generate spanning tree BPDUs, such as switches, or bridges and routers that bridge. If the switch receives a spanning tree BPDU on a port that has spanning tree PortFast and spanning tree BPDU guard enabled, the switch places the port in errdisabled mode to guard against potential loops. PortFast assumes that a port on a switch cannot generate a physical loop. Therefore, PortFast skips the initial spanning tree checks for that port, which avoids the timeout of end stations at bootup. The network administrator must carefully implement PortFast. On ports that have PortFast enabled, BPDU guard helps ensure the LAN stays loop-free.

---

This example shows how to turn on this feature. This example was chosen because creation of an errdisable situation is easy in this case:

<#root>

```
cat6k(config-if)#
```

**spanning-tree bpduguard enable**

---

In this example, a Catalyst 6509 switch is connected to another switch (a 6509). The 6500 sends BPDUs every 2 seconds (with use of the default spanning tree settings). When you enable PortFast on the 6509 switch port, the BPDU guard feature watches for BPDUs that come in on this port. When a BPDU comes into the port (a device that is not an end device is detected on that port), the BPDU guard feature errdisable the port to avoid the possibility of a spanning tree loop.

<#root>

```
cat6k(config-if)#
```

**spanning-tree portfast enable**

```
Warning: Spantree port fast start can only be enabled on ports connected
to a single host.  Connecting hubs, concentrators, switches, bridges, etc. to
a fast start port can cause temporary spanning tree loops.

%PM-SP-4-ERR_DISABLE: bpduguard error detected on Gi4/1, putting Gi4/1 in err-disable state.
```

---

In this message, the switch indicates it received a BPDU on a PortFast-enabled port, and the switch shuts down port Gi4/1.

<#root>

```
cat6k#

show interfaces gigabitethernet 4/1 status


Port    Name               Status       Vlan       Duplex  Speed Type

Gi4/1                      err-disabled 100          full   1000 1000BaseSX
```

---

Because this port connects to another switch, disable PortFast and the interface level BPDU Guard configuration, then reenable the port after you verify the intended spanning-tree topology. The connection is improper because PortFast and BPDU Guard are enabled, and the switch connects to another switch. Remember, PortFast is only for use on ports that connect to end stations.

```
<#root>

cat6k(config-if)#

spanning-tree portfast disable

cat6k(config-if)#

no spanning-tree bpduguard enable
```

---

- UDLD:
    - The UDLD protocol allows devices that are connected through fiber-optic or copper Ethernet cables (such as, Category 5 cabling) to monitor the physical configuration of the cables and detect when a unidirectional link exists. When a unidirectional link is detected, UDLD shuts down the affected port and alerts the user. Unidirectional links can cause a variety of problems, which include spanning-tree topology loops.

---

**Note**: UDLD exchanges protocol packets between the neighboring devices. Both devices on the link must support UDLD and have UDLD enabled on the respective ports. If you have UDLD enabled on only one port of a link, it can also leave the end configured with UDLD to go to errdisable state.

---

Each switch port configured for UDLD sends UDLD protocol packets that contain the port device (or port ID) and the neighbor device (or port IDs) that are seen by UDLD on that port. The neighboring ports must see their own device or port ID (echo) in the packets received from the other side. If the port does not see its own device or port ID in the incoming UDLD packets for a specific duration of time, the link is considered unidirectional. Therefore, the respective port is disabled and a message that is similar to this, is printed on the console:

```
PM-SP-4-ERR_DISABLE: udld error detected on Gi4/1, putting Gi4/1 in err-disable state.
```

---

- Link-Flap Error:
    - Link flap means the interface continually moves up and down. In this example, the link-flap threshold is 5 flaps in 10 seconds; run the **show errdisable flap-values** command for the threshold on the affected switch. The common cause of link flap is a Layer 1 issue such as a bad cable, duplex mismatch, or bad Gigabit Interface Converter (GBIC) card. Look at the console messages or the messages that were sent to the syslog server that state the reason for the port shutdown.

```
%PM-4-ERR_DISABLE: link-flap error detected on Gi4/1, putting Gi4/1 in err-disable state
```

---

Run this command to view the flap values:

<#root>

```
cat6k#
```

**show errdisable flap-values**

```
ErrDisable Reason    Flaps    Time (sec)
-----------------    ------   ----------
pagp-flap              3       30
dtp-flap               3       30
link-flap              5       10
```

---

- Loopback Error:
    - A loopback error occurs when the keepalive packet is looped back to the port that sent the keepalive. The switch sends keepalives to all interfaces by default. A device can loop the packets back to the source interface, which usually occurs because there is a logical loop in the network that the spanning tree has not blocked. The source interface receives the keepalive packet that it sent out, and the switch disables the interface (errdisable). This message occurs because the keepalive packet is looped back to the port that sent the keepalive:

```
%PM-4-ERR_DISABLE: loopback error detected on Gi4/1, putting Gi4/1 in err-disable state
```

---

Keepalives are sent on all interfaces by default in Cisco IOS® Software Release 12.1EA-based software. In Cisco IOS® Software Release 12.2SE-based software and later, keepalives are not sent by default on fiber

and uplink interfaces.

Investigate the loopback condition first. For an affected release and interface, consider a software upgrade; disable keepalives only after you assess the loss of loopback detection.

- Port Security Violation:
    - You can use port security with dynamically learned and static MAC addresses to restrict the ingress traffic of a port. To restrict the traffic, you can limit the MAC addresses that are allowed to send traffic into the port. In order to configure the switch port to enter the err-disabled state if there is a security violation, run this command:

<#root>

```
cat6k(config-if)#
```

```
switchport port-security violation shutdown
```

---

A security violation occurs in either of these two situations:

- When the maximum number of secure MAC addresses is reached on a secure port and the source MAC address of the ingress traffic differs from any of the identified secure MAC addresses. In this case, port security applies the configured violation mode.
- If traffic with a secure MAC address is configured or learned on one secure port attempts to access another secure port in the same VLAN. In this case, port security applies the shutdown violation mode.

---

- L2pt Guard:
    - When the Layer 2 PDUs enter the tunnel or access port on the inbound edge switch, the switch overwrites the original PDU-destination MAC address with a Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If 802.1Q tunneling is enabled, packets are double-tagged. The outer tag is the metro tag and the inner tag is the VLAN tag. The core switches ignore the inner tags and sends the packet to all trunk ports in the same metro VLAN. The edge switches on the outbound side restore the proper Layer 2 protocol and MAC address information and sends the packets to all tunnel or access ports in the same metro VLAN. Therefore, the Layer 2 PDUs are kept intact and delivered across the service-provider infrastructure to the other side of the network.

<#root>

```
Switch(config)#
```

```
interface gigabitethernet 0/7
```

```
Switch(config-if)#

l2protocol-tunnel {cdp | vtp | stp}
```

---

The interface moves to errdisabled state. If an encapsulated PDU (with the proprietary destination MAC address) is received from a tunnel port or access port with Layer 2 tunneling enabled, the tunnel port is shut down to prevent loops. The port also shuts down when a configured shutdown threshold for the protocol is reached. You can manually reenable the port (run a **shutdown**, **no shutdown** command sequence) or if errdisable recovery is enabled, the operation is retried after a specified time interval.

---

To enable timed automatic recovery for ports disabled by L2pt Guard, configure **errdisable recovery cause l2ptguard**. After you correct the underlying problem, run the interface **shutdown** and **no shutdown** commands for manual recovery. This command is used to configure the recovery mechanism from a Layer 2 maximum rate error so the interface can be moved out of the disabled state and tried again. You can also set the time interval. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.

- Incorrect SFP Cable:
    - Ports change into errdisable state with the "`%PHY-4-SFP_NOT_SUPPORTED`" error message when you connect Catalyst 3560 and Catalyst 3750 Switches and use an SFP Interconnect Cable.
    - The Cisco Catalyst 3560 SFP Interconnect Cable (CAB-SFP-50CM=) provides a low-cost, point-to-point, Gigabit Ethernet connection between Catalyst 3560 Series Switches. The 50-centimeter (cm) cable is an alternative to the SFP transceivers to interconnect Catalyst 3560 Series Switches through their SFP ports over a short distance. Check the hardware compatibility information for the specific Catalyst 3560 model before you use the SFP Interconnect Cable.
    - When a Catalyst 3560 Switch is connected to a Catalyst 3750 or any other type of Catalyst switch model, you cannot use the CAB-SFP-50CM= cable. You can connect both switches with a copper cable and a compatible copper SFP, such as GLC-T when supported, at each end instead of a CAB-SFP-50CM= cable.

---

- 802.1X Security Violation:

```
DOT1X-SP-5-SECURITY_VIOLATION: Security violation on interface GigabitEthernet4/8, New MAC address
%PM-SP-4-ERR_DISABLE: security-violation error detected on Gi4/8, putting Gi4/8 in err-disable sta
```

  This message indicates the port on the specified interface is configured in single-host mode. Any new host detected on the interface is treated as a security violation. The port has been err-disabled.

---

- Ensure only one host is connected to the port. If you need to connect an IP phone and a host behind it

to the port, configure Multidomain Authentication Mode on that switchport.

- The Multidomain authentication (MDA) mode allows an IP phone and a single host behind the IP phone to authenticate independently, with 802.1X, MAC authentication bypass (MAB), or (for the host only) web-based authentication. In this application, Multidomain refers to two domains — data and voice — and only two MAC addresses are allowed per port. The switch can place the host in the data VLAN and the IP phone in the voice VLAN, though they appear to be on the same switch port. The data VLAN assignment can be obtained from the vendor-specific attributes (VSAs) received from the AAA server within authentication.

- For more information, refer to the IEEE 802.1X Multidomain Authentication document.

- Reenable the Errdisabled Ports.

---

After you resolve the root problem, the ports are still disabled if you have not configured errdisable recovery on the switch. In this case, you must reenable the ports manually. To manually reenable an affected port after you correct the cause, enter interface configuration mode and run **shutdown**, then **no shutdown** commands.

The **errdisable recovery** command allows you to choose the type of errors that automatically reenable the ports after a specified amount of time. The **show errdisable recovery** command shows the default error-disable recovery state for all the possible conditions.

<#root>

cat6k#

**show errdisable recovery**

```
Recovery Status                     Timer Status
---------------                     ------------
udld                                  Disabled
bpduguard                             Disabled
security-violation                    Disabled
channel-misconfig                     Disabled
vmps                                  Disabled
pagp-flap                             Disabled
dtp-flap                              Disabled
link-flap                             Disabled
l2ptguard                             Disabled
psecure-violation                     Disabled
gbic-invalid                          Disabled
dhcp-rate-limit                       Disabled
mac-limit                             Disabled
unicast-flood                         Disabled
storm-control                         Disabled
arp-inspection                        Disabled
loopback                              Disabled
link-monitor-failure                  Disabled
oam-remote-failure critical-event     Disabled
```

```
oam-remote-failure dying-gasp           Disabled
oam-remote-failure link-fault           Disabled
dot1ad-incomp-etype                     Not supported
dot1ad-incomp-tunnel                    Not supported
mvrp                                    Not supported
transceiver-incomp                      Not supported
VSL transceiver-incomp                  Not supported
packet-buffer                           Not supported
FEX Licensing module removed            Not supported
inline-power                            Not supported

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

cat6k#
```

---



**Note**: The default timeout interval is 300 seconds and, by default, the timeout feature is disabled.

---

To turn on **errdisable recovery** and choose the errdisable conditions, run this command:

<#root>

cat6k#

**configure terminal**

cat6k(config)#

**errdisable recovery cause ?**

```
  all                   Enable timer to recover from all causes
  arp-inspection        Enable timer to recover from arp inspection error
                        disable state
  bpduguard             Enable timer to recover from BPDU Guard error disable
                        state
  channel-misconfig     Enable timer to recover from channel misconfig disable
                        state
  dhcp-rate-limit       Enable timer to recover from dhcp-rate-limit error
                        disable state
  dtp-flap              Enable timer to recover from dtp-flap error disable
                        state
  gbic-invalid          Enable timer to recover from invalid GBIC error disable
                        state
  l2ptguard             Enable timer to recover from l2protocol-tunnel error
                        disable state
  link-flap             Enable timer to recover from link-flap error disable
                        state
  link-monitor-failure  Enable timer to recover from link monitoring failure
  loopback              Enable timer to recover from loopback disable state
  mac-limit             Enable timer to recover from mac limit disable state
  oam-remote-failure    Enable timer to recover from remote failure detected by
                        OAM
  pagp-flap             Enable timer to recover from pagp-flap error disable
```

```
                        state
  psecure-violation     Enable timer to recover from psecure violation disable
                        state
  security-violation    Enable timer to recover from 802.1x violation disable
                        state
  storm-control         Enable timer to recover from storm-control error
                        disable state
  udld                  Enable timer to recover from udld error disable state
  unicast-flood         Enable timer to recover from unicast flood disable
                        state
  vmps                  Enable timer to recover from vmps shutdown error
                        disable state
```

---

This example shows how to enable the BPDU guard errdisable recovery condition:

<#root>

cat6k(config)#

**errdisable recovery cause bpduguard**

cat6k(config)#

**end**

---

A nice feature of this command is if you enable errdisable recovery, it lists general reasons the ports have been put into the err-disabled state. In this example, notice the BPDU guard feature was the reason for the shutdown of port 2/4:

<#root>

cat6k#

**show errdisable recovery**

```
Recovery Status                    Timer Status
---------------                    ------------
udld                                 Disabled
```

**bpduguard Enabled**

```
security-violation                   Disabled
channel-misconfig                    Disabled
vmps                                 Disabled
pagp-flap                            Disabled
```

```
dtp-flap                                Disabled
link-flap                               Disabled
l2ptguard                               Disabled
psecure-violation                       Disabled
gbic-invalid                            Disabled
dhcp-rate-limit                         Disabled
mac-limit                               Disabled
unicast-flood                           Disabled
storm-control                           Disabled
arp-inspection                          Disabled
loopback                                Disabled
link-monitor-failure                    Disabled
oam-remote-failure critical-event       Disabled
oam-remote-failure dying-gasp           Disabled
oam-remote-failure link-fault           Disabled
dot1ad-incomp-etype                     Not supported
dot1ad-incomp-tunnel                    Not supported
mvrp                                    Not supported
transceiver-incomp                      Not supported
VSL transceiver-incomp                  Not supported
packet-buffer                           Not supported
FEX Licensing module removed            Not supported
inline-power                            Not supported


Timer interval: 300 seconds


Interfaces that will be enabled at the next timeout:

Interface      Errdisable reason      Time left(sec)
---------    ---------------------    --------------
  Fa2/4                bpduguard          290
```

---

If an errdisable recovery condition is enabled, ports disabled for that cause are retried after the configured interval. If the condition persists, the ports can become err-disabled again. You can change this default of 300 seconds by running the **errdisable recovery interval <timer_interval_in_seconds>** command under the global configuration. This next example changes the errdisable recovery interval from 300 to 400 seconds:

<#root>

cat6k#

**configure terminal**

```
Enter configuration commands, one per line.  End with CNTL/Z.
cat6k(config)#
```

**errdisable recovery interval 400**

```
cat6k(config)#
```

```
end

cat6k#

show errdisable recovery


Recovery Status                       Timer Status
---------------                       ------------
udld                                    Disabled
bpduguard                               Disabled
security-violation                      Disabled
channel-misconfig                       Disabled
vmps                                    Disabled
pagp-flap                               Disabled
dtp-flap                                Disabled
link-flap                               Disabled
l2ptguard                               Disabled
psecure-violation                       Disabled
gbic-invalid                            Disabled
dhcp-rate-limit                         Disabled
mac-limit                               Disabled
unicast-flood                           Disabled
storm-control                           Disabled
arp-inspection                          Disabled
loopback                                Disabled
link-monitor-failure                    Disabled
oam-remote-failure critical-event       Disabled
oam-remote-failure dying-gasp           Disabled
oam-remote-failure link-fault           Disabled
dot1ad-incomp-etype                     Not supported
dot1ad-incomp-tunnel                    Not supported
mvrp                                    Not supported
transceiver-incomp                      Not supported
VSL transceiver-incomp                  Not supported
packet-buffer                           Not supported
FEX Licensing module removed            Not supported
inline-power                            Not supported


Timer interval: 400 seconds


Interfaces that will be enabled at the next timeout:

cat6k#
```

---

# Verify

- **show version**: Displays the version of the software that is used on the switch.
- **show interfaces <interface> status**: Shows the current status of the switch port. Helps to confirm the port is no longer err-disabled.

- **show errdisable detect**: Displays errdisable detection settings. Run **show errdisable recovery** to inspect recovery settings and listed recovery candidates.

---

# Troubleshoot

- **show interfaces status err-disabled**: Shows which local ports are involved in the errdisabled state.
- **show etherchannel summary**: Shows the current status of the EtherChannel.
- **show errdisable recovery**: Shows enabled recovery causes, the configured interval, and interfaces listed for the next retry.
- **show errdisable detect**: Shows the reason for the errdisable status.

---

# Related Information

- Troubleshoot Hardware and Issues on Catalyst 6500/6000 Switches
- Understand the Spanning Tree PortFast BPDU Guard Enhancement
- Understand EtherChannel Inconsistency Detection
- Troubleshoot Switch Port and Interface Problems