Cisco – Troubleshooting WS–X6348 Module Port Connectivity for Catalyst 6500/6000 Using CatOS (Partner Version)

Document ID: 29060

Contents

Introduction

This document discusses detailed troubleshooting for the WS–X6348 module on the Catalyst 6500/6000 running CatOS.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- Catalyst 6500 with Supervisor II with Multilayer Switch Feature Card 2 (MSFC2)
- WS–X6348 module
- CatOS version 6.3.9

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, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Background Information

Coil and Pinnacle Architecture

Each WS−X6348 card has a single Pinnacle Application−Specific Integrated Circuit (ASIC) that connects the module to both the 32 GB data bus backplane of the switch, as well as to four separate Coil ASICs on the same module through a single Gigabit connection to each. Each of the four Coil ASICs connects to 12 10/100 ports on the front panel of the module. This list provides more information on the connections:

- Ports 1 through 12 use Coil 1, which connects to Pinnacle port 1.
- Ports 13 through 24 use Coil 2, which connects to Pinnacle port 2.
- Ports 25 through 36 use Coil 3, which connects to Pinnacle port 3.
- Finally, ports 37 through 48 use Coil 4, which connects to Pinnacle port 4 on the module.

An understanding of this architecture is important as it can help in troubleshooting port problems. For example, if a group of 12 10/100 ports fails the online diagnostics, this typically indicates a Coil ASIC failure or Pinnacle port failure. See step 22 in order to learn more about the **show test** *module#* command.

Known Issues

1. Cisco bug ID CSCdu03935 (registered customers only) : 6348−RJ−45 Coil Pinnacle Header Checksum Error

   You see this error message:

   ```
   %SYS-5-SYS_LCPERR5:Module 9: Coil Pinnacle Header Checksum Error - Port #37
   ```

   If you see only the previous message and no other Coil−related messages in the syslogs or in the output of the **show logging buff 1023** command, and the transmit is stuck on one port, not a group of 12 ports, complete these steps in order to fix the problem:

   a. Disable and enable the ports.
   b. Soft−reset the module. Issue the **reset** *<module#>* command.
   c. Hard−reset the module. Issue the **set module power up|down** *<module#>* command.

   If after you complete step a and/or b and/or c, the card comes online and all the ports pass diagnostics, which is shown when you issue the **show test <module#>** command, and traffic starts to pass fine, you most likely experience Cisco bug ID CSCdu03935 (registered customers only) . The fix is in these CatOS releases and later:

   - 5.5(18)
   - 6.3(10)
   - 7.4(3)

2. You see a message similar to one or more of these error messages in the syslogs or the **show logging buff 1023** command output:

   - Coil Pinnacle Header Checksum
   - Coil Mdtif State Machine Error
   - Coil Mdtif Packet CRC Error
   - Coil Pb Rx Underflow Error
   - Coil Pb Rx Parity Error

   If you see one or more of these messages, and you have a group of 12 ports stuck and does not pass traffic, complete these steps:

a. Disable and enable the ports.
b. Soft−reset the module. Issue the **reset** *<module#>* command.
c. Hard−reset the module. Issue the **set module power up|down** *<module#>* command.

After complete steps b and/or c, contact the Cisco Technical Support with the previous information if you encounter one or more of these problems:

♦ The module does not come online.
♦ The module comes online, but a group of 12 ports fails diagnostics, which is seen in the output from the **show test** *<module#>* command.
♦ The module is stuck in the other state when it boots up.
♦ All port LEDs on the module become amber.
♦ All ports are in the `err-disabled` state as seen when you issue the **show** *<module#>* command.

Summary of Commands

This is a list of commands that are used in order to troubleshoot the WS−X6348 module connectivity issues in this document.

- **show module** *<module#>*
- **show config** *<module#>***.**
- **show logging buffer 1023**
- **show cam dynamic** *<module#/port>*
- **show trunk** *<module#/port>*
- **show spantree** *<module#/port>*
- **show cdp neighbor** *<module#/port>* **detail**

  Get three snapshots of each of these commands in order to monitor counter increments, for steps 8 through 19 only.
- **show port** *<module#/port>*
- **show mac** *<module#/port>*
- **show counters** *<module#/port>*
- **show intcounters** *<module#/port>* (Introduced in CatOS release 5.5(12), 6.3(4), and 7.x.)

  show log <module#>
- **show asicreg** *<module#/port>* **pinnacle errcounters**
- **show asicreg** *<module#/port>* **pinnacle pointers**
- **show asicreg** *<module#/port>* **pinnacle all**
- **show asicreg** *<module#/port>* **coil errcounters**
- **show asicreg** *<module#/port>* **coil pointers**
- **show asicreg** *<module#/port>* **coil 129**
- **show asicreg** *<module#/port>* **coil all**
- **show asicreg** *<module#/port>* **mii_phy all**

  Note: This Command Line Interface (CLI) does not currently work from CatOS release 6.3(8) and later. Refer to Cisco Bug ID CSCdz26435 (registered customers only) for more information.
- **show ltl** *<module#/port>*
- **show cbl** *<module#>*
- **set test diag complete**

  reset *<module#>*

  show test *<module#>*

Troubleshooting the Catalyst 6500/6000 WS−X6348 Module Port Connectivity

These are the steps to perform port connectivity troubleshooting on the Catalyst 6500/6000 WS−X6348 module.

Step−by−step Instructions

Complete these steps:

1. Check the software version in use and make sure there are no known WS−X6348 issues with that code. Verify the module is a WS−X6348 and that the status is `ok`.

```
esc-6509-c (enable) show module 6
Mod Slot Ports Module-Type                Model               Sub Status
--- ---- ----- ------------------------   -------------------- --- --------
6   6    48    10/100BaseTX Ethernet      WS-X6348-RJ-45       no  ok

Mod Module-Name          Serial-Num
--- -------------------  -----------
6                        SAD04170FPY

Mod MAC-Address(es)                         Hw     Fw         Sw
--- -------------------------------------   ------ ---------- -----------------
6   00-01-97-15-03-a0 to 00-01-97-15-03-cf 1.1    5.3(1)     6.3(9)
esc-6509-c (enable)
```

   In the previous command output, check the status of the module. It can be in one of these four states:

   ♦ OK Everything is fine.
   ♦ power−deny Not enough power is available to power the module.
   ♦ other Most likely the Serial Communication Protocol (SCP) communication is broken.
   ♦ faulty/unknown This indicates most likely a bad module or slot.
   ♦ err−disabled View the output from the **show log** command, which is shown in step 3, in order to see if there are any messages on why the module is in the err−disabled state.

2. Verify that the configuration for the module and its ports is correct. Make sure that options such as the **set port host** command, are enabled when appropriate.

```
esc-6509-c (enable) show config 6
This command shows non-default configurations only.
Use 'show config  all' to show both default and non-default configurations.
....................
begin
!
# ***** NON-DEFAULT CONFIGURATION *****
!
!
#time: Sun Oct 20 2002, 12:17:49
!
# default port status is enable
!
!
#module 6 : 48-port 10/100BaseTX Ethernet
set vlan 175  6/1-2
end
esc-6509-c (enable)
```

3. Issue the **show logging buff 1023** command in order to check for any port−related error messages in the log.

The output for this command is intentionally not shown as it is specific to each switch.

4. Verify that dynamic Content Addressable Memory (CAM) entries are created for any traffic that enters the port you are troubleshooting. Make sure that the CAM entry is associated with the correct VLAN.

```
esc-6509-c (enable) show cam dynamic 6/1
* = Static Entry. + = Permanent Entry. # = System Entry. R = Router Entry.
X = Port Security Entry $ = Dot1x Security Entry

VLAN  Dest MAC/Route Des    [CoS]  Destination Ports or VCs / [Protocol Type]
----  -----------------     -----  ------------------------------------------
175   00-d0-06-26-f4-00            6/1 [ALL]
175   00-e0-1e-a4-88-af            6/1 [ALL]
175   00-90-6d-fb-88-00            6/1 [ALL]
175   08-00-2b-2f-f4-dc            6/1 [ALL]
175   aa-00-04-00-01-a4            6/1 [ALL]
175   08-00-2b-2f-f3-b4            6/1 [ALL]
175   00-00-0c-0b-f8-98            6/1 [ALL]
175   00-00-0c-ff-ec-c9            6/1 [ALL]
175   00-03-e3-48-a6-e0            6/1 [ALL]
175   00-05-74-19-59-8a            6/1 [ALL]
175   00-08-e2-c3-60-a8            6/1 [ALL]
175   00-50-54-7c-f2-e0            6/1 [ALL]
175   00-50-54-75-dd-74            6/1 [ALL]
175   00-50-0b-6c-b8-00            6/1 [ALL]
175   00-04-5a-6c-6a-3a            6/1 [ALL]
175   00-00-0c-34-7b-16            6/1 [ALL]
175   00-00-0c-0c-19-36            6/1 [ALL]
175   08-00-69-07-b1-c8            6/1 [ALL]
Total Matching CAM Entries Displayed  =18
esc-6509-c (enable)
```

5. If a port is configured as a trunk, check to make sure it is in the correct status and that the appropriate VLANs are spanning–tree forwarding and not pruned by VLAN Trunk Protocol (VTP). For a dot1q trunk, also make sure that the native VLAN matches that of the device on the other side of the trunk.

```
esc-6509-e> (enable) show trunk 3/1
* - indicates vtp domain mismatch
Port      Mode          Encapsulation  Status        Native vlan
--------  -----------   -------------  ------------  -----------
 3/1      desirable     dot1q          trunking      1

Port      Vlans allowed on trunk
--------  ---------------------------------------------------------------------
 3/1      1-1005,1025-4094

Port      Vlans allowed and active in management domain
--------  ---------------------------------------------------------------------
 3/1      1-50,79-81,175-176,997-999

Port      Vlans in spanning tree forwarding state and not pruned
--------  ---------------------------------------------------------------------
 3/1      1-50,79-81,175-176,997-999
esc-6509-e> (enable)
```

6. Make sure that the port in question is forwarding for spanning–tree on the correct VLAN. Also, that portfast is enabled or disabled where appropriate.

```
esc-6509-c (enable) show spantree 6/1
Port                    Vlan Port-State    Cost      Prio Portfast Channel_id
----------------------- ---- ------------- --------- ---- -------- ----------
 6/1                    175  forwarding              19   32 disabled 0
esc-6509-c (enable)
```

7. If the port is connected to another Cisco device, use Cisco Discovery Protocol (CDP) in order to check if the port can see the device.

Note: CDP must be enabled on the switch and the other Cisco device. Also note that CDP is Cisco proprietary, and does not work with non−Cisco devices.

```
esc-6509-c (enable) show cdp port 6/1
CDP              : enabled
Message Interval : 60
Hold Time        : 180
Version          : V2
Device Id Format : Other


Port      CDP Status
--------  ----------
 6/1      enabled
esc-6509-c (enable)
```

In this example, port 6/1 on the Catalyst 6509 switch connects to Fast Ethernet interface 0/4 on a Catalyst 3500XL.

```
esc-6509-c (enable) show cdp neighbor 6/1 detail
Port (Our Port): 6/1
Device-ID: esc-cat3500xl-1
Device Addresses:
  IP Address: 172.16.176.200
Holdtime: 150 sec
Capabilities: TRANSPARENT_BRIDGE SWITCH
Version:
  Cisco Internetwork Operating System Software
  IOS (tm) C3500XL Software (C3500XL-C3H2S-M), Version 12.0(5.1)XW, MAINTENANCEE
  Copyright (c) 1986-2000 by cisco Systems, Inc.
  Compiled Thu 21-Dec-00 12:04 by devgoyal
Platform: cisco WS-C3548-XL
Port-ID (Port on Neighbors's Device): FastEthernet0/4
VTP Management Domain: sj-et
Native VLAN: unknown
Duplex: unknown
System Name: unknown
System Object ID: unknown
Management Addresses: unknown
Physical Location: unknown
esc-6509-c (enable)
```

Since CDP is Cisco proprietary, care must be taken. CDP packets are sent to a well−known multicast destination MAC address 01−00−0C−CC−CC−CC. A Cisco switch not configured for CDP, or a non−Cisco switch, typically treat CDP packets like any multicast and flood them throughout the VLAN. If two Cisco switches with CDP enabled are connected through a non−CDP−capable switch, this can result in those two CDP−enabled switches thinking they are CDP neighbors when, in fact, there is actually another switch in between them. Keep this in mind when you troubleshoot.

8. Check the configuration, state, and health of the port in trouble. You can also issue the **show port** <*module#*> command if you want to look at all the ports for a given module.

```
esc-6509-c (enable) show port 6/1
Port  Name                 Status     Vlan       Duplex Speed Type
----- -------------------- ---------- ---------- ------ ----- ------------
 6/1                       connected  175        a-full a-100 10/100BaseTX


Port  AuxiliaryVlan AuxVlan-Status    InlinePowered    PowerAllocated
                                      Admin Oper   Detected mWatt mA @42V
----- ------------- -------------- ----- ------ -------- ----- --------
 6/1  none          none               -    -        -      -     -


Port  Security Violation Shutdown-Time Age-Time Max-Addr Trap     IfIndex
----- -------- --------- ------------- -------- -------- -------- -------
 6/1  disabled shutdown              0        0        1 disabled      99
```

```
Port  Num-Addr Secure-Src-Addr   Age-Left Last-Src-Addr      Shutdown/Time-Left
----- -------- ---------------- -------- ---------------- ------------------
 6/1         0                -        -                -       -        -

Port     Broadcast-Limit Multicast Unicast Total-Drop
-------- --------------- --------- ------- --------------------
 6/1                   -         -       -                    0

Port  Send FlowControl  Receive FlowControl   RxPause     TxPause
      admin    oper      admin     oper
----- -------- -------- --------- ---------   ---------- ----------
 6/1  off      off      off       off                 0          0

Port  Status      Channel              Admin Ch
                  Mode                 Group Id
----- ----------  -------------------- ----- -----
 6/1  connected   auto silent             34     0

Port  Align-Err  FCS-Err     Xmit-Err    Rcv-Err     UnderSize
----- ---------- ---------- ---------- ---------- ---------
 6/1           0          0          0          0         0

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

Port  Last-Time-Cleared
----- -------------------------
 6/1  Sun Oct 13 2002, 16:37:58
esc-6509-c (enable)
```

♦ `Status` Can display these states:

- ◊ `connected`
- ◊ `notconnect`
- ◊ `connecting`
- ◊ `standby`
- ◊ `faulty`
- ◊ `inactive`
- ◊ `shutdown`
- ◊ `disabled`
- ◊ `err-disabled`
- ◊ `monitor`
- ◊ `active`
- ◊ `dot1p`
- ◊ `untagged`
- ◊ `inactive`
- ◊ `onhook`

If a port is in the `notconnect` state, check the cabling as well as the device connected to the other end. If a port is in the `faulty` state, it indicates a hardware problem. Issue the **show test** <*module#*> command for module diagnostic results. If the port is in the `inactive` state, issue the **show vlan** command in order to make sure that the VLAN of the port still exists and issue the **set port enable** <*module#/port*> in order to try to re−enable the port. VTP problems can sometimes cause a VLAN to be deleted, which results in ports associated with that VLAN becoming inactive.

♦ `Vlan` This field displays trunk if it is a trunk port, or the VLAN number the port is a member of if it is an access port.

♦ `speed and duplex` These fields have an **a** in front of the value displayed, such as `a-full`, if the value was obtained through auto negotiation. If the port is hardcoded for

speed and duplex the a is not present. While not in a connected state, an auto–negotiation–enabled port displays auto in these fields. Make sure that the device attached to this port has the same settings as the port regarding either hard setting the speed and duplex or auto–negotiating the speed and duplex.

♦ If port security is enabled, make sure the appropriate MAC addresses are allowed to pass through the port, and that the port is not shut down due to a security violation.

♦ If broadcast suppression is enabled, check the number of dropped packets to make sure this is not the cause of traffic problems on the port.

♦ If flow control is enabled, ensure that the other side of the link supports flow control as well, and make sure that the settings match on both ends.

♦ If the port is configured as part of an EtherChannel, its state and the state of the other ports in the channel are displayed. Information on the neighbor device appears based on information obtained through CDP, if you assume that CDP is enabled on both devices in the channel.

♦ `FCS-Err` This is the number of valid size frames with Frame Check Sequence (FCS) errors but no framing errors. This is typically a physical issue, for example, cabling, a bad port, or a bad Network Interface Card (NIC), but can also indicate a duplex mismatch.

♦ `Align-Err` This is the number of frames with alignment errors, which are frames that do not end with an even number of octets and have a bad Cyclic Redundancy Check (CRC), received on the port. These usually indicate a physical problem, for example, cabling, a bad port, or a bad NIC, but can also indicate a duplex mismatch. When the cable is first connected to the port, some of these errors can occur. Also, if there is a hub connected to the port, collisions between other devices on the hub can cause these errors.

♦ `Xmit-Err and Rcv-Err` This indicates that the internal port transmit (Tx) and receive (Rx) buffers are full. A common cause of Xmit–Err is traffic from a high bandwidth link being switched to a lower bandwidth link, or traffic from multiple inbound links being switched to a single outbound link. For example, if a large amount of bursty traffic comes in on a gigabit port and is switched out to a 100 Mbps port, this can cause the `Xmit-Err` field to increment on the 100 Mbps port. This is because that output buffer of the port is overwhelmed by the excess traffic due to the speed mismatch between the incoming and outgoing bandwidths.

♦ `Late-coll` (late collisions) This is the number of times that a collision is detected on a particular port late in the transmission process. For a 10 Mbit/sec port, this is later than 512 bit–times into the transmission of a packet. Five hundred and twelve bit–times corresponds to 51.2 microseconds on a 10 Mbit/sec system. This error can indicate a duplex mismatch among other things. For the duplex mismatch scenario, the late collision are seen on the half duplex side. As the half duplex side transmits, the full duplex side does not wait its turn and transmits simultaneously causing a late collision. Late collisions can also indicate an Ethernet cable or segment that is too long. Collisions should not be seen on ports configured as full duplex.

♦ `Single-coll` (single collision) This is the number of times one collision occurs before the port transmits a frame to the media successfully. Collisions are normal for ports configured as half duplex, but should not be seen on full duplex ports. If collisions increase dramatically, this points to a highly utilized link or possibly a duplex mismatch with the attached device.

♦ `Multi-coll` (multiple collision) This is the number of times multiple collisions occur before the port transmits a frame to the media successfully. Collisions are normal for ports configured as half duplex, but should not be seen on full duplex ports. If collisions increase dramatically, this points to a highly utilized link or possibly a duplex mismatch with the attached device.

♦ `Excess-coll` (excessive collisions) This is a count of frames for which transmission on a particular port fails due to excessive collisions. An excessive collision happens when a packet has a collision 16 times in a row. The packet is then dropped. Excessive collisions are typically an indication that the load on the segment needs to be split across multiple segments, but can also point to a duplex mismatch with the attached device. Collisions should not be seen on ports configured as full duplex.

- ♦ `Carri-Sen` (carrier sense) This occurs every time an Ethernet controller wants to send data on a half duplex connection. The controller senses the wire and checks if it is not busy before transmitting. This is normal on a half duplex Ethernet segment.
- ♦ `Undersize` The frames received that are smaller than the minimum IEEE 802.3 frame size of 64 bytes long, which exclude framing bits, but include FCS octets, that are otherwise well formed and have a valid CRC. Check the device that sends out these frames.
- ♦ `Runts` The frames received that are smaller than the minimum IEEE 802.3 frame size (64 bytes for Ethernet), and with a bad CRC. This can be caused by a duplex mismatch and physical problems, such as a bad cable, port, or NIC on the attached device.
- ♦ `Giants` Frames that exceed the maximum IEEE 802.3 frame size (1518 bytes for non–jumbo Ethernet), and have a bad FCS. Try to find the offending device and remove it from the network. In many cases, it is the result of a bad NIC.
- ♦ Issue the **clear counters [all | mod/port]** command in order to reset the statistics for the **show port**, **show Mac**, and **show counters** commands.

  Refer to Quick Links to Catalyst 6500 Family Switch and ROM Monitor Commands for more information and a further explanation of the various fields in the **show port** command output.
9. Check that traffic counters increment both inbound and outbound on the port. You can also issue the **show Mac**<*module#*> command if you want to look at the MAC info for all ports for a given module.

```
esc-6509-c (enable) show Mac 6/1

Port     Rcv-Unicast              Rcv-Multicast            Rcv-Broadcast
-------- --------------------     --------------------     --------------------
 6/1                    20890                   894039                    74883

Port     Xmit-Unicast             Xmit-Multicast           Xmit-Broadcast
-------- --------------------     --------------------     --------------------
 6/1                    12845                    73660                      179

Port     Rcv-Octet                Xmit-Octet
-------- --------------------     --------------------
 6/1                 79498714                  8738501

MAC      Dely-Exced MTU-Exced  In-Discard Out-Discard
-------- ---------- ---------- ---------- -----------
 6/1              0          0          0           0

Port  Last-Time-Cleared
----- -------------------------
 6/1  Sun Oct 13 2002, 16:37:58
esc-6509-c (enable)
```

- ♦ The previous output shows the total unicast, multicast, and broadcast packets received (Rcv) and transmitted (Xmit) on a port.

  Note: If the port is an Inter–Switch Link Protocol (ISL) trunk, all traffic are multicast and all ISL headers use the destination multicast address 01–00–0C–CC–CC–CC.
- ♦ `Dely-Exced` This is thee number of frames discarded by this port due to an excessive transmit delay through the switch. This counter should never go up unless the port is under very high utilization.
- ♦ `MTU Exceed` This is an indication that one of the devices on that port or segment is transmitting more than the allowed frame size (1518 bytes for non–jumbo Ethernet).
- ♦ `In-Discard` This is the result of incoming valid frames that were discarded because the frame did not need to be switched. This can be normal if a hub is connected to a port and two devices on that hub exchange data. The switch port still sees the data but does not have to switch it, since the CAM table shows the MAC address of both devices associated with the same port, and so it is discarded. This counter can also increment on a port configured as a

trunk if that trunk is blocking for some VLANs, or on a port that is the only member of a VLAN.

♦ `Out-Discard` This is the number of outbound packets chosen to be discarded even though no packet errors have been detected. One possible reason to discard such a packet can be to free up buffer space.

♦ Issue the **clear counters [all | mod/port]** command in order to reset the statistics for the **show port**, **show Mac**, and **show counters** commands.

Refer to Quick Links to Catalyst 6500 Family Switch and ROM Monitor Commands for more information and a further explanation of the various fields in the **show Mac** command output.

10. Check the detailed statistics for a specific port.

```
esc-6509-c (enable) show counters 6/1
64 bit counters
0   rxHCTotalPkts                        =               364517
1   txHCTotalPkts                        =                35104
2   rxHCUnicastPkts                      =                10281
3   txHCUnicastPkts                      =                 6678
4   rxHCMulticastPkts                    =               338957
5   txHCMulticastPkts                    =                28343
6   rxHCBroadcastPkts                    =                15279
7   txHCBroadcastPkts                    =                   83
8   rxHCOctets                           =             29291862
9   txHCOctets                           =              3460655
10  rxTxHCPkts64Octets                   =               181165
11  rxTxHCPkts65to127Octets              =               201314
12  rxTxHCPkts128to255Octets             =                 5546
13  rxTxHCPkts256to511Octets             =                11425
14  rxTxHCpkts512to1023Octets            =                   81
15  rxTxHCpkts1024to1518Octets           =                   89
16  txHCTrunkFrames                      =                    0
17  rxHCTrunkFrames                      =                    0
18  rxHCDropEvents                       =                    0
32 bit counters
0   rxCRCAlignErrors                     =          0
1   rxUndersizedPkts                     =          0
2   rxOversizedPkts                      =          0
3   rxFragmentPkts                       =          0
4   rxJabbers                            =          0
5   txCollisions                         =          0
6   ifInErrors                           =          0
7   ifOutErrors                          =          0
8   ifInDiscards                         =          0
9   ifInUnknownProtos                    =          0
10  ifOutDiscards                        =          0
11  txDelayExceededDiscards              =          0
12  txCRC                                =          0
13  linkChange                           =          4
14  wrongEncapFrames                     =          0
0   dot3StatsAlignmentErrors             =          0
1   dot3StatsFCSErrors                   =          0
2   dot3StatsSingleColFrames             =          0
3   dot3StatsMultiColFrames              =          0
4   dot3StatsSQETestErrors               =          0
5   dot3StatsDeferredTransmisions        =          0
6   dot3StatsLateCollisions              =          0
7   dot3StatsExcessiveCollisions         =          0
8   dot3StatsInternalMacTransmitErrors   =          0
9   dot3StatsCarrierSenseErrors          =          0
10  dot3StatsFrameTooLongs               =          0
11  dot3StatsInternalMacReceiveErrors    =          0
0   txPause                              =          0
1   rxPause                              =          0
0   rxTotalDrops                         =          0
```

```
1  rxFIFOFull                        =         0
2  rxBadCode                         =         0
Last-Time-Cleared
-------------------------
Sun Oct 20 2002, 16:23:06
esc-6509-c (enable)
```

This is a list of some of the non–generic counter details from the previous output :

- `RxFragmentPkts` This is the total number of packets received that do not end with an even number of octets (alignment error) or that has an FCS error, and is less than 64 octets in length. This excludes framing bits, but includes FCS octets.
- `dot3StatsInternalMacReceiveErrors` A count of frames for which reception on a particular port fails due to an internal MAC sublayer receive error. A frame is only counted if it is not counted by the corresponding instance of either `dot3StatsFrameTooLongs`, `dot3StatsAlignmentErrors`, or `dot3StatsFCSErrors`. In particular, an instance of this object can represent a count of receive errors on a particular port that are not otherwise counted.
- `dot3StatsInternalMacTransmitErrors` This is a count of frames for which transmission on a particular port fails due to an internal MAC sublayer transmit error. A frame is only counted if it is not counted by the corresponding instance of either `dot3StatsLateCollisions`, `dot3StatsExcessiveCollisions`, or `dot3StatsCarrierSenseErrors`.
- `RxJabbers` The total number of packets received that are longer than 1518 octets, which excludes framing bits, but includes FCS octets, and do not end with an even number of octets (alignment error), or had an FCS error. The recommended action is to isolate the device that sends out these packets.
- `txDelayExceededDiscards` The number of frames discarded by this port due to an excessive transmit delay through the switch. This counter is the same as the `Dely-Exced` counter in the output from the **show Mac** command, and should never go up unless the port is under very high utilization.
- `IfInUnknownProtos` The number of inbound packets with unknown protocols.
- `TxCRC` This increments when frames are transmitted with a bad CRC, but it does not include frames aborted due to a late collision. This counter typically increments on an egress port when transmitting a frame that is received as an ISL frame on an ingress port, but that carries an Ethernet packet with a bad CRC inside it, while the ISL packet itself has a good CRC. It can also be caused by bad switch hardware. A way to troubleshoot this is to send broadcast traffic on a port and see if the counter increments on all egress connected ports. If this happens independent of the port where you send traffic into, there is a failure in the switch hardware, most probably the chassis or supervisory module. If the counter increments only when a certain module is used to send traffic into, this module has a hardware failure. If the counter only increments on a few ports, the ports themselves have a problem. If the cause cannot be determined by the previous test, check the neighbor switches that are ISL connected, or check ISL connected end–devices. Contact Cisco Technical Support if you need further assistance.
- `dot3StatsSQETestErrors` This is a count of times that the `SQE TEST ERROR` message is generated by the physical layer signaling sublayer (PLS) for a particular interface. The `SQE TEST ERROR message` is defined in section 7.2.2.2.4 of the American National Standards Institute (ANSI)/IEEE 802.3–1985 and its generation is described in section 7.2.4.6 of the same document. This counter should never go up, since it is only of relevance to external Ethernet transceivers.
- `dot3StatsCarrierSenseErrors` This is the number of times that the carrier sense condition is lost or never asserted during an attempt to transmit a frame on a particular port. The count represented by an instance of this object is incremented at most once for each transmission attempt, even if the carrier sense condition fluctuates during a transmission

attempt. This counter is the same counter as the `Carri-Sen` field in the output of the **show port** command. This is normal on an half duplex Ethernet segment.

- ♦ `linkChange` This is the number of times the port toggles between a `connected` state to a `non-connected` state. If this counter increments constantly, it means there is something wrong with this port, the cable attached to this port, or the device at the other end of the cable.
- ♦ `dot3StatsFrameTooLongs` This is a count of frames received on a particular interface that exceeds the maximum permitted frame size. Check the device attached to the port.
- ♦ `dot3StatsFCSErrors` This is a count of valid frames received on a particular interface that end with an even number of octets but do not pass the FCS check. This is typically a physical issue, for example, cabling, bad port, or bad NIC card, but can also indicate a duplex mismatch. This is the same counter as the `FCS-Err` field in the output from the **show port** command.
- ♦ `dot3StatsSingleColFrames` This is a count of successfully transmitted frames on a particular port for which transmission is initially inhibited by exactly one collision. Collisions are normal for ports configured as half duplex, but should not be seen on full duplex ports. If collisions increase dramatically this points to a highly utilized link, or possibly a duplex mismatch with the attached device. This is the same counter as the `Single-Coll` field in the output from the **show port** command.
- ♦ `dot3StatsMultiColFrames` This is a count of successfully transmitted frames on a particular port for which transmission is initially inhibited by more than one collision. Collisions are normal for ports configured as half duplex, but should not be seen on full duplex ports. If collisions increase dramatically this points to a highly utilized link or possibly a duplex mismatch with the attached device. This is the same counter as the `Multi-Coll` field in the output from the **show port** command.
- ♦ `dot3StatsExcessiveCollisions` This is a count of frames for which transmission on a particular port fails due to excessive collisions. An excessive collision happens when a packet has a collision 16 times in a row. The packet is then dropped. Excessive collisions are typically an indication that the load on the segment needs to be split across multiple segments, but can also point to a duplex mismatch with the attached device. Collisions should not be seen on ports configured as full duplex. This is the same counter as the `Excess-Coll` field in the output from the **show port** command.
- ♦ `dot3StatsLateCollisions` This is the number of times that a collision is detected on a particular port late in the transmission process. For a 10 Mbit/sec port this is later than 512 bit–times into the transmission of a packet. 512 bit–times corresponds to 51.2 microseconds on a 10 Mbit/sec system. A late collision is also considered a generic collision for purposes of other collision–related statistics. This counter is the same as the `Late-Coll` field in the output from the **show port** command, and can indicate a duplex mismatch among other things. For the duplex mismatch scenario, the late collision is seen on the half duplex side. As the half duplex side transmits, the full duplex side does not wait its turn and transmits simultaneously, which causes a late collision. Late collisions can also indicate an Ethernet cable or segment that is too long. Collisions should not be seen on ports configured as full duplex.
- ♦ `dot3StatsDeferredTx` A count of frames for which the first transmission attempt on a particular port is delayed because the medium is busy. This count does not include frames involved in collisions. Deferred transmissions are normal in Ethernet, however, a high count might indicate a highly loaded segment.
- ♦ `rxBadCode` This is a count of received frames for which the preamble has a bad code. Check the device connected to the port.
- ♦ `IfInDiscards` This is a count of valid frames received which are discarded by the forwarding process of the switch. This is the same counter as the `In-Discard` field in the output from the **show Mac** command. You see this when you receive traffic on a trunk for a specific VLAN while the switch does not have any other ports on that VLAN. You also see this counter incrementing when the destination address of the packet is learned on the port the packet is received on, or when a port is configured as a trunk and that trunk is blocking for

VLANs.

- ♦ `rxUndersizedPkts` The total number of packets received that are less than 64 octets long, which excludes framing bits, but includes FCS octets, and are otherwise well formed. This counter is the same as the `Undersize` field in the output from the **show port** command. Check the device that sends out these frames.
- ♦ `RxOversizePkts` The total number of packets received that are longer than 1518 octets, which excludes framing bits, but includes FCS octets, and are otherwise well formed. Check the device connected to this port. This counter can increment when the device attached to the port has ISL encapsulation enabled, and the port itself does not. This counter also increments if you receive jumbo frames without the configuration of a jumbo support on the port.
- ♦ `dot3StatsAlignmentErrors` The total number of packets received that have a length, which excludes framing bits, but includes FCS octets, of between 64 and 1518 octets, inclusive, but do not end with an even number of octets and have a bad FCS. This is the same counter as the `Align-Err` field in the output from the **show port** command. These errors usually indicate a physical problem, for example, cabling, bad port, or bad NIC card, but can also indicate a duplex mismatch. When the cable is first connected to the port, some of these errors can occur. Also, if there is a hub connected to the port, collisions between other devices on the hub can cause these errors.
- ♦ `rxTotalDrops` This counter includes a sum of these counters:

  ◊ The number of bad packets because of a CRC error.
  ◊ A coding violation or sequence error.
  ◊ The number of Color Blocking Logic (CBL) blocking drops.
  ◊ The number of instances of invalid encapsulation.
  ◊ The number of broadcast suppression drops.
  ◊ The number of drops because the packet length is less than 64 or greater than 1518 bytes.

  CBL refers to the spanning–tree state of a particular VLAN (color) on the port in question. If the port is in a spanning–tree blocking state for a particular VLAN, it is normal to drop packets received on that port for that VLAN. See step 21 for more information on CBL.

11. Check for incrementing errors. Also, issue the **show logging buffer 1023** command, which is shown in step 3, which syslogs any of these errors that occur on a port. Some errors cause the module to be reset by firmware to recover. This command was introduced in CatOS release 5.5(12), 6.3(4), and 7.x.

```
esc-6509-c (enable) show intcounters 6/1
MasterInt       : 0
PbUnderflow     : 0
Parity          : 0
InternalParity  : 0
PacketCRC       : 0
MdtifErr        : 0
CpuifErr        : 0
PnclChksum      : 0
```

Issue the **show log** command in order to get the history of the module resets.

```
esc-6509-c (enable) show log 6

Module 6 Log:
  Reset Count:   73
  Reset History: Sun Oct 13 2002, 15:51:18
                 Sun Oct 13 2002, 08:44:51
                 Sat Oct 12 2002, 22:48:11
                 Fri Oct 11 2002, 23:47:30
```

12. This command displays the registers of the Pinnacle ASIC that relates specifically to error counts. They should all be clean of errors. Take three snapshots in order to check for increments in the counters.

```
esc-6509-c (enable) show asicreg 6/1 pinnacle errcounters
00C5: PI_CI_S_HDR_FCS_REG                           = 0000
00C6: PI_CI_S_RBUS_FCS_REG                          = 0000
00C7: PI_CI_S_PKTCRC_ERR_REG                        = 0000
00C8: PI_CI_S_PKTLEN_ERR_REG                        = 0000
00C9: PI_CI_S_BPDU_OUTLOST_REG                      = 0000
00CE: PI_CI_S_HOLD_REG                              = 0000
00CA: PI_CI_S_QOS0_OUTLOST_REG                      = 0000
00CE: PI_CI_S_HOLD_REG                              = 0000
00CB: PI_CI_S_QOS1_OUTLOST_REG                      = 0000
00CE: PI_CI_S_HOLD_REG                              = 0000
00CC: PI_CI_S_QOS2_OUTLOST_REG                      = 0000
00CE: PI_CI_S_HOLD_REG                              = 0000
00CD: PI_CI_S_QOS3_OUTLOST_REG                      = 0000
00CE: PI_CI_S_HOLD_REG                              = 0000
0150: PI_GM_S_TX_PARERR_REG                         = 0000
0151: PI_GM_S_RX_PARERR_REG                         = 0000
0152: PI_GM_S_INCRC_ERR_REG                         = 0000
0153: PI_GM_S_CBL_DROP_REG                          = 0000
0154: PI_GM_S_TOTAL_DROP_REG                        = 0000
0158: PI_PN_S_CRC_ERR_CNT_REG                       = 0000
0159: PI_PN_S_RBUS_ERR_CNT_REG                      = 0000
015A: PI_PBT_S_BPDU_OUTLOST_REG                     = 0000
015F: PI_PBT_S_HOLD_REG                             = 0000
--More
```

<output truncated>

13. This command displays the pointer registers of the Pinnacle ASIC. Take three snapshots in order to check for changes in the counters to make sure the registers are not stuck.

```
esc-6509-c (enable) show asicreg 6/1 pinnacle pointers
003F: PI_INT_HI_WR_PTR_REG                          = 02DB
0040: PI_INT_HI_CMT_PTR_REG                         = 02DB
0041: PI_INT_HI_RD_PTR_REG                          = 02DB
0042: PI_INT_HI_DN_PTR_REG                          = 02DB
0044: PI_INT_LO_WR_PTR_REG                          = 04CC
0045: PI_INT_LO_CMT_PTR_REG                         = 04CC
0046: PI_INT_LO_RD_PTR_REG                          = 04CC
0047: PI_INT_LO_DN_PTR_REG                          = 04CC
010A: PI_PBT_HI_WR_PTR_MSB_REG                      = 0000
010B: PI_PBT_HI_WR_PTR15_0_REG                      = A94C
010C: PI_PBT_HI_CMT_PTR_MSB_REG                     = 0000
010D: PI_PBT_HI_CMT_PTR15_0_REG                     = A94B
010E: PI_PBT_HI_RD_PTR_MSB_REG                      = 0000
010F: PI_PBT_HI_RD_PTR15_0_REG                      = A94C
0112: PI_PBT_LO_WR_PTR_MSB_REG                      = 0000
0113: PI_PBT_LO_WR_PTR15_0_REG                      = CECC
0114: PI_PBT_LO_CMT_PTR_MSB_REG                     = 0000
0115: PI_PBT_LO_CMT_PTR15_0_REG                     = CECB
0116: PI_PBT_LO_RD_PTR_MSB_REG                      = 0000
0117: PI_PBT_LO_RD_PTR15_0_REG                      = CECC
011C: PI_PBR_WR_PTR_MSB_REG                         = 0000
011D: PI_PBR_WR_PTR15_0_REG                         = FA81
011E: PI_PBR_CMT_PTR_MSB_REG                        = 0000
011F: PI_PBR_CMT_PTR15_0_REG                        = FA7F
0120: PI_PBR_RD_PTR_MSB_REG                         = 0000
0121: PI_PBR_RD_PTR15_0_REG                         = FA80
0127: PI2_PBR_HI_WR_PTR_MSB                         = 0000
0128: PI2_PBR_HI_WR_PTR15_0                         = F672
0129: PI2_PBR_HI_CMT_PTR_MSB                        = 0000
012A: PI2_PBR_HI_CMT_PTR15_0                        = F670
012B: PI2_PBR_HI_RD_PTR_MSB                         = 0000
012C: PI2_PBR_HI_RD_PTR15_0                         = F671
013C: PI2_PBT_VHI_WR_PTR_MSB                        = 0000
013D: PI2_PBT_VHI_WR_PTR15_0                        = A58F
```

```
013E: PI2_PBT_VHI_CMT_PTR_MSB                     = 0000
013F: PI2_PBT_VHI_CMT_PTR15_0                     = A58E
0140: PI2_PBT_VHI_RD_PTR_MSB                      = 0000
0141: PI2_PBT_VHI_RD_PTR15_0                      = A58F
0142: PI2_PBT_VHI_FREE_CNT_MSB                    = 0000
0143: PI2_PBT_VHI_FREE_CNT15_0                    = 0400

esc-6509-c (enable)
```

- ♦ Pointers to internal packet buffers should move (PI_INT_HI& and PI_INT_LO& counters)
- ♦ Priority TX pointers to external packet buffers should move (PI_PBT_HI& and PI_PBT_LO& counters)
- ♦ Priority Rx pointers to external packet buffers should move (PI_PBR_HI& and PI_PBR_LO& counters)

14. Issue this command in order to dump all the Pinnacle ASIC register settings. Collect three snapshots of this in case it is requested by the TAC engineer.

```
esc-6509-c (enable) show asicreg 6/1 pinnacle all
0001: PI_CP_RESET0_1_REG                          = 1F1F
0002: PI_CP_RESET2_3_REG                          = 1F1F
0003: PI2_MII_PHY_ADDR                            = 0000
0004: PI2_MII_MGMT_ADDR                           = 0000
0005: PI2_MII_MGMT_CMD_STATUS                     = 0000
0006: PI2_MII_MGMT_DATA                           = 0000
0007: PI_CP_RESET_GEN_REG                         = 0000
0008: PI_CP_DISABLE0_3_REG                        = 0000
0009: PI_CP_CFG_REG                               = 1000
000A: PI_CP_PORT_NUM_REG                          = 0003
000B: PI_MATCH1_ADDR47_32_REG                     = 0100
000C: PI_MATCH1_ADDR31_16_REG                     = 0CCC
000D: PI_MATCH1_ADDR15_0_REG                      = CCCD
000E: PI_MATCH2_ADDR47_32_REG                     = 0000
000F: PI_MATCH2_ADDR31_16_REG                     = 0000
0010: PI_MATCH2_ADDR15_0_REG                      = 0000
0011: PI_GM_BCAST_INT_CNTR31_16_REG               = 0000
0012: PI_GM_BCAST_INT_CNTR15_0_REG                = 0000
0014: PI_GM_FC_DA_47_32_REG                       = 0180
0015: PI_GM_FC_DA_31_16_REG                       = C200
0016: PI_GM_FC_DA_15_0_REG                        = 0001
0017: PI_GM_ISL_SA47_32_REG                       = F000
0018: PI_GM_ISL_SA31_16_REG                       = 0000
--More
```

15. This command displays the registers of the Coil ASIC for the port that relates specifically to error counts. They should all be clean of errors. Take three snapshots in order to check for increments in the counters.

```
esc-6509-c (enable) show asicreg 6/1 coil errcounters
00C8: CO_PTX_S_DROP_CNT                           = 0000
00C9: CO_PTX_S_CRC0_CNT                           = 0000
00CA: CO_PRX_S_BAD_CNT                            = 0000
00CB: CO_PRX_S_ASSERT_FC                          = 0000
00CC: CO_PTX_S_ASSERT_FC                          = 0000
00CD: CO_PBR_ERR_COUNT                            = 0000
00CE: CO_PBT_ERR_COUNT                            = 0000
00CF: CO_PBR_FULL_DROP_COUNT                      = 0000
00D0: CO_PBT_FULL_DROP_COUNT                      = 0000
0153: CO_PRX_S_CBL_DROP                           = 0000
0154: CO_PRX_WRONG_ENCAP                          = 0000
0159: CO_PBT_S_BPDU_OUTLOST                       = 0000
015A: CO_PBT_S_QOS3_OUTLOST                       = 0000
```

```
015B: CO_PBT_S_QOS2_OUTLOST                      = 0000
015C: CO_PBT_S_QOS1_OUTLOST                      = 0000
015D: CO_PBT_S_QOS0_OUTLOST                      = 0000
015E: CO_PBR_S_BPDU_INLOST                       = 0000
015F: CO_PBR_S_QOS3_INLOST                       = 0000
0160: CO_PBR_S_QOS2_INLOST                       = 0000
0161: CO_PBR_S_QOS1_INLOST                       = 0000
0162: CO_PBR_S_QOS0_INLOST                       = 0000
016F: CO_PTX_S_CBL_DROP                          = 0000
0170: CO_PTX_S_CAP0_CNT                          = 0000
--More
```

<output truncated>

- ♦ The `CO_PRX_S_ASSERT_FC` and `CO_PTX_S_ASSERT_FC` counters can increment at times, which means there is congestion between the Pinnacle ASIC and the Coil ASIC associated with this port. These counters indicate that either the Coil ASIC receives flow control asserts from the Pinnacle ASIC, or sends flow control asserts to the Pinnacle ASIC through the gigabit connection between the ASICs. For example, if the Pinnacle receives a flow control assert from the Coil, it can mean that traffic comes into the Coil ASIC from the gigabit connection to the Pinnacle ASIC overwhelms the output buffers on one or more of the 12 10/100 ports associated with that Coil ASIC due to the speed mismatch involved. The Coil is flow controlling the Pinnacle in order to signal it to slow down transmission in order to prevent this. The `Xmit-err` counter in the output from the **show port** command, which is shown in step 8, indicates if the output buffers on any of the 12 10/100 ports are overrun.

  Note: By default, flow control between the Pinnacle and Coil ASICs is disabled:

  ```
  esc-6509-c (enable) show option flowcontrol
  Option flowcontrol: disabled
  ```

16. This command displays the pointer registers of the Coil ASIC associated with the port. Take three snapshots to check for changes in the counters in order to make sure the registers are not stuck.

    ```
    esc-6509-c (enable) show asicreg 6/1 coil pointers
    010B: CO_PBT_HI_WR_PTR                       = 01A0
    010D: CO_PBT_HI_WRCMT_PTR                    = 01A0
    010F: CO_PBT_HI_RD_PTR                       = 01A0
    0111: CO_PBT_HI_FREE_CNT                     = 0580
    0113: CO_PBT_LO_WR_PTR                       = 0557
    0115: CO_PBT_LO_WRCMT_PTR                    = 0557
    0117: CO_PBT_LO_RD_PTR                       = 0557
    0119: CO_PBT_LO_FREE_CNT                     = 1680
    011D: CO_PBR_WR_PTR                          = 0258
    011F: CO_PBR_WRCMT_PTR                       = 0257
    0121: CO_PBR_RD_PTR                          = 0257
    0123: CO_PBR_FREE_CNT                        = 03FF

    esc-6509-c (enable)
    ```

    - ♦ High and low Tx counters should move (CO_PBT_HI& and CO_PBT_LO&)
    - ♦ Rx counters should move (CO_PBR&)

17. Issue this command in order to dump the specific MAC control setting of the Coil ASIC register associated with the port. This can be used to verify that the duplex setting in the output from the **show port** command is really set in the Coil ASIC, which is particularly helpfully in auto−negotiation troubleshooting, whether jumbo packets are enabled in the ASIC for this port, which should match the setting seen in the output from the **show port jumbo** command, and that the MAC is not in loopback.

    ```
    esc-6509-c (enable) show asicreg 6/1 coil 129
    0129: CO_MAC_CONTROL1                        = 014C
    ```

```
esc-6509-c (enable)
```

This is the decode of the command output:

```
0x014C = 101001100 binary
Checking bit setting from right to left:
Bit5 = 0 (MAC loopback is disabled)
Bit6 = 0 (tx & rx of jumbo packets is disabled)
Bit7 = 1 (full duplex)
```

18. Issue this command to dump all the Coil ASIC register settings associated with the port. Collect three snapshots of this in case it is requested by the TAC engineer.

```
esc-6509-c (enable) show asicreg 6/1 coil all
0001: CO_TFIFO_CONFIG                          = 0001
0002: CO_CPU_DISABLE0_3                        = 0000
0003: CO_CPU_DISABLE4_7                        = 0000
0004: CO_CPU_DISABLE8_11                       = 0000
0005: CO_CPU_RESET_GEN                         = 0000
0006: CO_PORT_NUM                              = 0000
0007: CO_PB_CONFIG                             = 0000
0008: CO_CPU_MATCHA_ADDR47_32                  = 0180
0009: CO_CPU_MATCHA_ADDR31_16                  = C200
000A: CO_CPU_MATCHA_ADDR15_0                   = 0020
000B: CO_CPU_MATCHB_ADDR47_32                  = 0100
000C: CO_CPU_MATCHB_ADDR31_16                  = 0CCC
000D: CO_CPU_MATCHB_ADDR15_0                   = CCCD
000E: CO_CPU_MATCHC_ADDR47_32                  = 0000
000F: CO_CPU_MATCHC_ADDR31_16                  = 0000
0010: CO_CPU_MATCHC_ADDR15_0                   = 0000
0011: CO_MDT_CONFIG                            = 0000
0012: CO_MDR_BCAST_INT_CNTR15_0                = BEBC
0013: CO_MDR_FC_TYPE                           = 8808
0014: CO_MDR_FC_DA_47_32                       = 0180
0015: CO_MDR_FC_DA_31_16                       = C200
0016: CO_MDR_FC_DA_15_0                        = 0001
0017: CO_MDT_ISL_SA47_32                       = 0001
--More
```

<output truncated>

19. Issue this command in order to dump the media independent interface (mii) phy register settings associated with the port. Collect three snapshots of this in case it is requested by the TAC engineer. You can also decode register `0000`, `0001`, and `0005` in order to verify auto–negotiation settings for the port as shown here.

Note: This CLI is currently not functional from CatOS release 6.3(8) and later. Refer to Cisco bug ID CSCdz26435 (registered customers only) for more information..

```
esc-6509-e> (enable) show asicreg 2/1 mii_phy all
0000:                                          = 1000
0001:                                          = 782D
0002:                                          = 0040
0003:                                          = 6136
0004:                                          = 01E1
0005:                                          = 41E1
0006:                                          = 0003
0007:                                          = 0000
0008:                                          = 0000
0009:                                          = 0000
000A:                                          = 0000
000B:                                          = 0000
000C:                                          = 0000
000D:                                          = 0000
```

```
       000E:                                                = 0000
       000F:                                                = 0000
       0010:                                                = 5000
       0011:                                                = 0301
       0012:                                                = 0000
       0013:                                                = 0000
       0014:                                                = 0000
       0015:                                                = 02BA
       0016:                                                = 0F00
       --More
```

<output truncated>

The `mii_phy` settings for register `0000`, `0001`, and `0005` can be useful in order to help verify auto−negotiation settings.

- ♦ Registers `0000` and `0001` What is assumed for which the port is set.
- ♦ Register `0005` What is assumed the link partner (other end) is capable of, through auto−negotiation.

Key for register `0000`:

Bit	Description
15	Reset bit
14	Looback bit
13	Select 100M mode
12	Select Auto−Nego mode
11	Power down bit − ignore
10	Isolate mode
9	Restart Auto−Negotion
8	Full−Duplex Auto−off, 10 BT
7	Collision test
0 − 6	undefined

From the sample output register, `0000` = 1000 hex = 0001 0000 0000 0000 in binary. If you count from right to left (bit 0 to 15), and use the previous key, you can see that the only bit that is set to equal 1 is bit 12, which means it is assumed that our port is set to auto−negotiate, which can be verified with the **show port** command

```
       esc-6509-e (enable) show port 2/1
       Port  Name                 Status     Vlan       Duplex Speed Type
       ----- -------------------- ---------- ---------- ------ ----- ------------
        2/1                       connected  176        a-full a-100 10/100BaseTX
```

Key for register `0001`: (The port settings)

Bit	Description
15	100BaseT4 capable
14	100BaseTX, full duplex capable
13	100BaseTX, half duplex capable

Bit	Description
12	10BaseT, full duplex capable
11	10BaseT, half duplex capable
6 – 10	Undefined
5	Auto–Nego process completed
4	Remote fault
3	Auto configuration ability
2	Link status
1	Jabber detect
0	Extended capability

From the sample output register, `0001` = 782D hex = 0111 1000 0010 1101 in binary. If you count from right to left (bit 0 to 15), and use the previous key, you can see that the only bits set to equal 1 are 0,2,3,5 and 11 through 14. This means that you should have told your link partner that you support 10BaseT and 100BaseT in both half or full duplex mode through the auto–negotiation process. It also means that the auto–negotiation process is complete and that you have the link.

Key for register `0005`: (Link partner capability):

Bit	Description
15	Next page indicatin
14	Acknowledge
13	Remote fault
10 – 12	Undefined
9	100BT4 capable
8	100BTX full duplex
7	100BTX capable
6	10BT full duplex
5	10BTX capable

From the sample output register, `0005` = 41E1 hex = 0100 0001 1110 0001 in binary. If you count from right to left (bit 0 to 15), and use the previous key, you can see that the only bits 0, 5 through 8, and 14 are set to equal 1. This means that the device attached to this port has acknowledged through the auto–negotiation process that it supports both 10BT and 100BT as well as full duplex mode, since bits 5 through 8 are set as well as bit 14. Our switch port should agree to the best supported setting the attached device is capable of, which is 100/full in this case.

20. Check the Local Target Logic (LTL) setting of a port. LTL is used by the Supervisor in order to target a particular packet to the appropriate port. For example, if the Supervisor must forward a broadcast packet to all ports in a given VLAN, a particular LTL value is used in the result sent on the Result BUS (RBUS) in order to signal this to the line cards. If the broadcast does not get through on a port that it should be, check the LTL for that port. The same concept can be used for unicast packet, and unknown unicast flood problems. Before you look at LTL, verify that the ports are configured as they should be through the commands listed in the Summary of Commands section. Some bugs that relate to LTL problems in the past have involved the Switched Port Analyzer (SPAN) feature, since SPAN modifies the LTL so that a packet analyzer gets a copy of the traffic as well. Keep this in mind when

you troubleshoot.

```
esc-6509-c (enable) show ltl 6/1
Getting LTL Data from Module 6, for Port 1 enabled entries (0x0000 to 0xFFFF)
LTL memory bits work with active low (enabled with 0)
Valid Ports ->0x000F     0xFFFF FFFF FFFF
INDEX          LTL-A      LTL-B --------->
0x0140:        0xFFFE     0xFFFF FFFF FFFE
0x80AF:        0xFFFE     0xFFFF FFFF FFFC
0xC0AF:        0xFFFE     0xFFFF FFFF FFFC
```

LTL Details

- ♦ `0x0140` Software unicast LTL index
- ♦ `0x80` Hardware flood LTL index
- ♦ `0xC0` Hardware broadcast LTL index
- ♦ LTL−A is used by the Pinnacle (four gigabit ports per chip) ASIC, LTL−B is used by the Coil (12 10/100 ports per chip) ASICs.
- ♦ The index value of 0x0140 is for software unicast handling. This value is derived from the actual module and port number. 0x0140 = 0000 0001 0100 0000. But, only the last 10 bits (01 0100 0000) are used. For port 6/1, the port number minus one should equal the six least significant bits of the index (port 1   1 = 0dec = 000000). The module number minus one should be represented by the four most significant bits (module 6   1 = 5dec = 0101). If you put this module and port value together, it gives 01 0100 0000.
- ♦ The actual LTL−A and LTL−B value for the 0x0140 index is 0xFFFE and 0xFF&FFFE. If you convert this to binary (0xFFFE = 1111 1111 1111 1110) and read from right (port 1) to left, only port 1 is set with a value of 0 for both LTL−A and LTL−B. LTL−B represents the four Coil ASICs, so this means that the 0x0140 LTL index is used to send unicast traffic to port 6/1 only. LTL−A represents the four Pinnacle ports. Since port 6/1 is associated with Coil 1 (which handles ports 6/1 through 12) and Coil 1 connects to port 1 on the Pinnacle, port 1 of the Pinnacle is also set. A decode of the LTL index value for software unicast should only have the port in question (6/1) listed, since a unicast should only go out one port, and you have specified 6/1 in the **show ltl 6/1** command.
- ♦ The index values of 0x80 and 0xC0 are for hardware flood and broadcast. AF is the VLAN (0xAF = 175 decimal = VLAN 175). Unlike the software unicast LTL index which is specific to port 6/1, the broadcast and flood LTL indexes cover all ports for the whole module for the given VLAN.
- ♦ If you convert the Coil ASICs LTL−B value (0xFF&FFFC) of index 0x80AF and 0xC0AF to binary, it gives 0xFF&FFFC = 11&1111 1111 1111 1100. If you read from right (port 1) to left, only ports 1 and 2 are set with a value of 0, and so only 6/1 and 6/2 can forward unknown unicasts and broadcasts for VLAN 175 on module 6. If you issue the **show port** and/or the **show trunk** commands, this should reveal that 6/1 and 6/2 are the only active ports in VLAN 175 on module 6.

  Note: LTL should be set for a port even if it is in a spanning−tree blocked state.
- ♦ The conversion of the Pinnacle ASICs LTL−A value (0xFFFE) of index 0x80AF and 0xC0AF to binary gives 0xFFFE = 1111 1111 1111 1110. If you read from right (port 1) to left, only port 1 is set with a value of 0, and so only port 1 on the Pinnacle can forward unknown unicasts and broadcasts for VLAN 175 on module 6. Remember that each Coil ASIC handles 12 10/100 ports, so port 6/1 and 6/2 is part of the same Coil ASIC (the first Coil ASIC), which connects to port 1 of the Pinnacle. If a port associated with the second Coil ASIC on module 6 (ports 6/13 through 24) was also active in VLAN 175, that Coil ASIC corresponds to port 2 on the Pinnacle, and LTL−A is set to 0xFFFC = 1111 1100.

21. Check the CBL of a port. The color refers to the VLAN, so this command is used to verify the spanning−tree state of a given VLAN for a specific port. This can be used in order to verify that the

values seen in the output from the **show spantree** *<mod/port>* are actually set correctly in the Pinnacle and Coil ASICs.

```
esc-6509-c (enable) show cbl 6 af 5
Getting CBL Data from Module 6, Address 0x00AF, Length 5
CBL States(binary): 00-disabled, 01-Blocking/Listening, 10-Learning, 11-Forwardg
Word Index  ->     0        5    4    3    2    1    0
Valid Ports ->0x  0F     0x  FF   FF   FF   FF   FF   FF
VLAN          CBL-A      CBL-B ----------------------->
0x00AF:        0x0003     0x0000 0000 0000 0000 0000 0007
0x00B0:        0x0000     0x0000 0000 0000 0000 0000 0000
0x00B1:        0x0000     0x0000 0000 0000 0000 0000 0000
0x00B2:        0x0000     0x0000 0000 0000 0000 0000 0000
0x00B3:        0x0000     0x0000 0000 0000 0000 0000 0000
esc-6509-c (enable)
```

CBL Details

- The command syntax is **show cbl [module] [start vlan (in hex)] [length]**, where length is how many VLANs to display info about starting at the start vlan, which is the vlan number from which your output starts. The default length is 1 if not specified. For example, the output from the **show cbl 6 af 5** command displays CBL info for module 6 starts with VLAN 0xAF = 175 dec and includes the next four VLANs after that (VLAN 176 to 179), due to the length field that is set to 5.
- Unlike LTL, CBL requires two bits to represent each port since there are more variables to be represented, for example, 00 = disabled, 01 = blocking/listening, 10 = learning, and 11 = forwarding.
- In the example in this document the CBL setting means: what spanning−tree settings are made for each of the module 6 ports in VLANs 175 through 179.
- If you concentrate on VLAN 175 (0xAF), there is a CBL−A value of 0x0003. CBL−A is for the Pinnacle ASIC. If you convert this to binary, it gives 0x0003 = 0000 0000 0000 0011. If you read from right (port 1) to left, with two bits to represent each port, port 1 is set to 11 = forwarding, while all other ports are set to 00 = disabled for spanning−tree. Pinnacle port 1 corresponds to Coil 1 which controls the first 12 10/100 ports on the module (6/1 through 12). This means that one or more ports in the range of 6/1 through 12 must be in a spanning−tree forwarding state, and that ports in the range of 6/13 through 48 must not be. Check the Coil ASIC settings in CBL−B in order to confirm this.
- If you concentrate on VLAN 175 (0xAF), there is a CBL−B value of 0x00&0007. CBL−B is for the Coil ASICs. If you convert this to binary, it gives 0x00&0007 = 0000&.0000 0000 0000 0111. If you read from right (port 1) to left, with two bits to represent each port, port 1 is set to 11 = forwarding, port 2 is set to 01 = blocking/listening, while all other ports are set to 00 = disabled for spanning−tree on module 6, VLAN 175. In this case 6/1 and 6/2 are the only active module 6 ports that are members of VLAN 175, and so the other ports show up as disable. The output from the **show spantree [vlan]** or **show spantree [mod/port]** can be used in order to verify that the CBL is set correctly.

```
esc-6509-c (enable) show spantree 175
VLAN 175
Spanning tree mode        PVST+
Spanning tree type        ieee
Spanning tree enabled

Designated Root           00-30-94-93-e5-80
Designated Root Priority  1
Designated Root Cost      76
Designated Root Port      6/1
Root Max Age   20 sec   Hello Time 2  sec   Forward Delay 15 sec

Bridge ID MAC ADDR        00-d0-02-ea-1c-ae
Bridge ID Priority        32768
```

```
       Bridge Max Age 20 sec    Hello Time 2  sec    Forward Delay 15 sec

       Port                     Vlan Port-State    Cost      Prio Portfast Channel_id
       ------------------------ ---- ------------- --------- ---- -------- ----------
         3/1                     175  forwarding           4   32 disabled 0
         6/1                     175  forwarding          19   32 disabled 0
         6/2                     175  blocking           100   32 disabled 0
        16/1                     175  forwarding           4   32 enabled  0
       esc-6509-c (enable)
```

22. Issue the **show test** *<module#>* command in order to check the results of the online diagnostic test performed at switch boot time or when a module is reset. The results of these tests can be used to determine if a hardware component failure is detected on the module. It is important to set the diagnostic mode to complete, otherwise all or some of the diagnostic tests are skipped. If a hardware component failure has occurred between now and the last switch or module reset, the diagnostics must be run again through a switch or module reset in order to detect the failure.

    Complete these steps in order to run the diagnostic tests for a module:

    a. Set the diagnostic mode to `complete`.

    ```
    esc-6509-c (enable) set test diag complete
    Diagnostic level set to complete.
    ```

    b. Reset the module.

    ```
    esc-6509-c (enable) reset 6

    This command will reset module 6 and may disconnect your telnet session.

    Do you want to continue (y/n) [n]? y
    ```

    c. View the diagnostic test result for the ports on the module for any indication of a failure. Also check for failures in groups of 12 ports, which suggest a Coil ASIC failure or Pinnacle port failure.

    ```
    esc-6509-c (enable) show test 6

    Diagnostic mode: complete   (mode at next reset: complete)

    Module 6 : 48-port 10/100BaseTX Ethernet

    Line Card Status for Module 6 : PASS

    Port Status :
      Ports 1   2   3   4   5   6   7   8   9  10 11 12 13 14 15 16 17 18 19 20 21 22 23
      -------------------------------------------------------------------------------
            .   .   .   .   .   .   .   .   .   .  .  .  .  .  .  .  .  .  .  .  .  .  .
            25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
            -------------------------------------------------------------------------
            .   .   .   .   .   .   .   .   .   .  .  .  .  .  .  .  .  .  .  .  .  .  .

    Line Card Diag Status for Module 6  (. = Pass, F = Fail, N = N/A)

     Loopback Status [Reported by Module 2] :
      Ports 1   2   3   4   5   6   7   8   9  10 11 12 13 14 15 16 17 18 19 20 21 22 23
      -------------------------------------------------------------------------------
            .   .   .   .   .   .   .   .   .   .  .  .  .  .  .  .  .  .  .  .  .  .  .
      Ports 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
      -------------------------------------------------------------------------------
            .   .   .   .   .   .   .   .   .   .  .  .  .  .  .  .  .  .  .  .  .  .  .

     InlineRewrite Status :
      Ports 1   2   3   4   5   6   7   8   9  10 11 12 13 14 15 16 17 18 19 20 21 22 23
      -------------------------------------------------------------------------------
            .   .   .   .   .   .   .   .   .   .  .  .  .  .  .  .  .  .  .  .  .  .  .
    ```

```
Ports 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
-----------------------------------------------------------------------
       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

esc-6509-c (enable)
```

Related Information

- **Troubleshooting Catalyst 6500/6000 Series Switches Running CatOS on the Supervisor Engine and Cisco IOS on the MSFC**
- **Troubleshooting Hardware and Related Issues on the MSFC, MSFC2, and MSFC2a**
- **LAN Switches Hardware Support**
- **Technical Support & Documentation – Cisco Systems**