Guide Contents
Troubleshooting Cisco IOS Voice Overview
Debug Command Output on Cisco IOS Voice Gateways
Filtering Troubleshooting Output
Cisco VoIP Internal Error Codes
Troubleshooting Cisco IOS Voice Telephony
Troubleshooting Cisco IOS Voice Protocols
Troubleshooting Cisco IOS Telephony Applications
Monitoring the Cisco IOS Voice Network
Cause Codes and Debug Values

Contents

DSP Symptoms

Digital signal processors (DSPs) enable Cisco platforms to efficiently process digital voice traffic. The following symptoms can be attributed to DSP hardware or software issues:

- No audio heard by either party or one-way audio on the voice path after the call is connected.
- Call setup failure, such as the inability to detect or transmit proper Channel Associated Signaling (CAS) state transitions.
- Channels are stuck in the PARK state and cannot be used.
- Error messages on the console or in the router log complain of DSP timeouts.

Voice DSP Control Message Logger

This section contains the following information:

- Message Logger Overview
- Configuration Tasks
- Configuration Examples

Caution: Using the logger feature in a production network environment increases CPU and memory usage on the gateway.

Note: We recommend that you work closely with your Cisco representative to use this feature. If you are experiencing problems with certain voice calls, the engineering team at Cisco might ask you to capture the control messages using the voice DSP logger. You can capture these messages by turning on the logger, repeating the problematic calls, and capturing the logs. Only Cisco engineers can determine if you should send the logs in for further review.

Message Logger Overview

The Voice DSP Control Message Logger feature provides improved debugging capabilities through Cisco IOS software and allows loggin of control messages that pass through the voice DSP firmware on the host port interface (HPI). The logged messages can later be examined for diagnosis of voice problems.

There are two main types of HPI messages that flow through the HPI interface: control messages and data messages. Control messages carry control information between Cisco IOS software and the DSP. Data messages carry voice data.

The Voice DSP Contol Message Logger feature captures control messages sent between the platform-independent portions of Cisco IOS software and the DSP. The HPI subsystem that is in Cisco IOS software contains the platform-independent portion of Cisco IOS software. This feature addresses the

sequence and contents of the control messages. The logged messages can be checked for parameters that might cause undesirable DSP behavior including the following:

- Incorrect parameters
- Out-of-sequence function calls
- Interactions between parameters of different HPI calls

In many cases, DSP problems have been the result of bad control messages. By logging all of these messages for offline analysis, you can better integrate and debug at-speed issues for analysis.

Message Capture

Message capture occurs when voice control messages are captured and passed between the Cisco IOS software and the DSP to a ring buffer. Some of these messages are sent in fast-path routines that run at a high priority, so the capture of the message must be done as quickly as possible. After the fast-path routine messages have been sent, a normal priority process sends the messages that are waiting in the ring buffer to off-router data storage through the Cisco IOS File System (IFS).

The size of the ring buffer is configurable through the use of the **voice hpi capture** command. If the ring buffer fills up faster than the normal priority process can move the messages off the router, some of the control messages are dropped.

Counters keep track of the number of messages that are waiting in the ring buffer, the number of messages that are sent, and the number of messages that are dropped. When message capture is enabled and a message arrives for which there is no buffer space, a missed-message count is started. The next time there is room for a message on the ring, the dropped-message count is included with the message data. This alerts the software that processes the messages to the missed messages, and it provides data capture feedback that helps you configure the ring buffer size to your specifications.

If messages are dropped during the capture, the ability to check the messages becomes limited. A complete capture is required for analysis.

Benefits

Improved DSP Reliability

This feature improves the reliability of DSPs by improving debug capabilities. Unexpected sequences of calls or parameters that cause DSP problems are difficult to debug because many calls can be made to the DSP before any ill effects are noticed. Systems that are running under load are more likely to encounter subtle timing-related issues that occur infrequently and are very hard to reproduce and debug. These parameters are marked as bad in HPI calls to the DSP, and they can cause undesirable DSP behavior. There fore, the logger intercepts those parameters that pass between Cisco IOS software and the DSP that can later be checked for errors.

Robust Firmware

This feature makes the T1-based DSP firmware more robust, adding debug capabilities and enabling better field support.

Restrictions

The Voice DSP Contol Message Logger feature is supported only on systems that use the HPI interface.

Configuration Tasks

See the following sections for configuration tasks for the Voice DSP Control Message Logger feature. Each task in the list is identified as either required or optional.

- Configuring the Voice DSP Control Message Logger (required)
- Verifying the Voice DSP Control Message Logger (optional)

Configuring the Voice DSP Control Message Logger

You can start the message logger by choosing the amount of memory (greater than 324 bytes) that the buffer-queueing system can allocate to the free message pool. HPI messages are captured until buffer space runs out. Once the buffer-queueing system is running, the transport process attempts to connect to a new or existing capture destination URL. A version message is written to the URL, and if the version message is accepted, any messages placed into the message queue are written to the URL. If a new URL is entered using command-line interface (CLI), an open URL is closed, and the system tries to write to the new URL. If the new URL fails, the transport process exits. The transport process is restarted when another URL is entered or the system is restarted.

To configure the message logger, use the following commands beginning in privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show voice hpi capture**
3. **debug hpi capture**
4. **configure terminal**
5. **voice hpi capture buffer** *size*
6. **voice hpi capture destination** *url*
7. **exit**
8. **show voice hpi capture**
9. **configure terminal**
10. **no voice hpi capture buffer 0**
11. **exit**
12. **show voice hpi capture**

DETAILED STEPS

	Command	Purpose
1.	**enable** Example: `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
2.	**show voice hpi capture** Example: `Router# show voice hpi capture`	(Optional) Displays the capture status and statistics. • Use this command to confirm logger status and examine the logger status output when the

		logger is running.
3.	**debug hpi capture** Example: `Router# debug hpi capture`	(Optional) Turns on the debug output for the logger. • It is recommended that you enable the debug output for the logger when you are interacting with it using CLI.
4.	**configure** {**terminal** | **memory** | **network**} Example: `Router# configure terminal`	Enters global configuration mode.
5.	**voice hpi capture buffer** *size* Example: `Router(config)# voice hpi capture buffer 122`	(Optional if you already have a nonzero buffer) Allocates the buffer for storing captured messages. • Starts the logger by giving it a nonzero buffer size. • The **no** form of the command turns the logger off by setting the buffer size to zero. Valid range is from 0 to 9000000. • If the buffer overflows so that messages are dropped during the capture, the buffer size needs to be increased and the capture needs to be restarted. **Note**: To change buffer size, first configure buffer size to zero, and then set the buffer size to your specifications.
6.	**voice hpi capture destination** *url* Example: `Router(config)# voice hpi capture destination 172.14.33.255`	(Optional if you already have a destination or do not want to change the currently assigned destination) Sets up an FTP destination file to which the logged data is sent. • The url argument is the destination address.
7.	**exit** Example: `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
8.	**show voice hpi capture** Example: `Router# show voice hpi capture`	(Optional) Displays the capture status and statistics. • Use this command to confirm logger status and examine the logger status output when the

		logger is running. **Note**: At this point, you can execute voice calls and capture the control messages. You can capture these messages by turning on the logger, repeating problematic calls, and capturing the logs. Cisco engineers can determine if you should send the logs in for further review.
9.	**configure** {**terminal** | **memory** | **network**} Example: `Router# configure terminal`	Enters global configuration mode.
10.	**no voice hpi capture buffer 0** Example: `Router(config)# no voice hpi capture buffer 0`	(Optional if you already have a nonzero buffer) Turns the logger off by setting the buffer size to zero and stops message capture.
11.	**exit** Example: `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
12.	**show voice hpi capture** Example: `Router# show voice hpi capture`	Verifies that no messages were dropped during the capture. **Note**: At this point, gather the captured messages in the destination files on your PC or UNIX station and send the information to Cisco TAC for analysis.

Verifying the Voice DSP Control Message Logger

To verify and print capture status and statistics, use the **show voice hpi capture** privileged EXEC command. This command displays the capture status and statistics and checks that the message counter is incrementing. If messages are being dropped consistently, try increasing the buffer size.

Note: If you want to stop the logger or change the buffer to another size, first set the buffer size to zero.

Troubleshooting Tips

Use the **debug hpi capture** command in privileged EXEC mode to turn on the debug output for the logger. Enable the debug output for the logger by using the CLI.

Configuration Examples

This section provides configuration examples for the Voice DSP Control Message Logger feature. This section contains the following examples:

- Starting the Logger Feature Example

Starting the Logger Feature Example

In the following example, the **voice hpi capture** buffer command is used in global configuration mode to start the logger by giving it a buffer size of 700000:

```
Router(config)# voice hpi capture buffer 700000
*Mar  1 00:24:47.090:caplog:caplog_cli_interface:hpi capture buffer size set to 700000  bytes
*Mar  1 00:24:47.090:caplog:caplog_logger_init:TRUE, Started task HPI Logger (PID 140
*Mar  1 00:24:47.150:caplog:caplog_cache_init:TRUE, malloc_named(699952), 2134 elements (each 328
*Mar  1 00:24:47.154:caplog:caplog_logger_proc:Terminating...
```

In the following example, the **show voice hpi capture** command is used in privileged EXEC mode to examine the logger status output now that the logger is enabled:

```
Router# show voice hpi capture
HPI Capture is on and is logging to URL <www.company.com>
0 messages sent to URL, 0 messages droppedMessage Buffer (total:inuse:free)  2134:0000:2134
Buffer Memory:699952 bytes, Message size:328 bytes
```

Setting up an FTP Destination Example

In the following example, the **voice hpi capture** destination command is used in global configuration mode to set up an FTP destination file where the logged data can be sent:

```
Router(config)# voice hpi capture destination ftp://100.00.100.200/d:\test_data.dat
*Mar  1 00:26:54.617:caplog:caplog_cli_interface:hpi capture destination:ftp://100.00.100.200/d:\t
*Mar  1 00:26:54.621:caplog:caplog_logger_init:TRUE, Started task HPI Logger (PID 140)
*Mar  1  00:26:54.621:caplog:caplog_logger_proc:Attempting to open ftp://100.00.100.200/d:\test_da
*Mar  1  00:26:57.091:caplog:caplog_logger_proc:Logging to ftp://100.00.100.200/d:\test_data.dat
```

In the following example, the **show voice hpi capture** command is used in privileged EXEC mode to examine the logger status output:

```
Router# show voice hpi capture
HPI Capture is on and is logging to URL ftp://172.23.184.216/d:\test_data.dat1
messages sent to URL, 0 messages dropped
Message Buffer (total:inuse:free)  2134:0000:2134
Buffer Memory:699952 bytes, Message size:328 bytes
```

Verifying Configuration Example

In the following example, the **show voice hpi capture** command is used in privileged EXEC mode to check on the status of the logger before, during, and after configuration:

```
Router# show voice hpi capture
HPI Capture is off and is logging to URL <no URL>
0 messages sent to URL, 0 messages dropped
Message Buffer (total:inuse:free)  0000:0000:0000
Buffer Memory:0 bytes, Message size:328 bytes
```

Voice Call Tuning

The Voice Call Tuning feature monitors the interface between Cisco IOS software and a system's digital signaling processors (DSPs) in real time and reports status on the following: packet flow, DSP state,

echo-cancellation state, and jitter state. The feature also allows you to manipulate echo-cancellation and jitter-buffer parameters in real time. For details on this feature, see the Voice Call Tuning article.

Voice DSP Crash Dump File Analysis

The Voice Crash Dump File Analysis feature allows Cisco IOS voice platforms using Texas Instruments DSPs the ability to capture the contents of the DSP memory into a file in the event of a DSP crash. By making this crash dump file available for offline analysis, engineers can better and more quickly determine and fix the cause of the crash.

DSP crash dump analysis allows to you do the following:

- Detect when control messages have been lost between Cisco IOS software and the DSP
- Detect when the DSP has crashed
- Collect an image of the DSP memory after a DSP crash and put it into a file for analysis later by an engineer

When these events have been detected, they are announced by console alarms. You can enable and disable this feature and specify where the crash dump is to be written using Cisco IOS command-line interface (CLI). The active part of the stack is written to the console, while the entire contents of the DSP memory is written to the crash dump file. You can request that a dump file be written into a "smart" slot 0 or slot 1 flash card, or sent to a server using TFTP or FTP, or it may be written directly to Flash.

How to Configure Voice DSP Crash Dump File Analysis

To configure Voice DSP crash dump file analysis, use the following steps:

SUMMARY STEPS

1. **enable**
2. **configure** terminal { memory | network}
3. **voice dsp crash-dump destination** url
4. **voice dsp crash-dump file-limit** limit-number
5. **exit**
6. **show voice dsp crash-dump**

DETAILED STEPS

	Command or Action	Purpose
1.	**enable** Example: `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
2.	**configure** {**terminal**	**memory**	**network**} Example: `Router# configure terminal`	Enters global configuration mode.
3.	**voice dsp crash-dump destination** url Example:	(Required) Designates a valid file system where crash dump analysis is stored.

	`Router(config)# voice dsp crash-dump destination 175.101.122`	• The *url* argument must be set to a valid file system. • The destination URL can be one of the following ♦ The file on a TFTP server with the following format: tftp://x.x.x.x/subfolder/filename. The x.x.x.x value is the IP address of the TFTP server • ♦ The file on the flashcard of the router, with the following format: slot0:filename **Note**: The DSP crash dump feature is disabled when the crash-dump destination is not specified.
4.	**voice dsp crash-dump file-limit** limit-number Example: `Router(config)# voice dsp crash-dump file-limit 99`	(Required) Sets the number of files you would like to write. • The crash dump **file-limit** keyword must be set to a non-zero value. The default is that the crash dump capability is turned off, as the url argument is empty, and the file-number argument is zero. • The limit-number argument can range from 0 to 99. **Note**: The DSP crash dump feature is disabled when the crash-dump file limit is set to 0.
5.	**exit** Example: `Router(config)# exit`	Exits to privileged EXEC mode.
6.	**show voice dsp crash-dump** Example: `Router# show voice dsp crash-dump`	(Optional) Displays voice DSP crash dump information

Troubleshooting Voice DSP Crash Dump File Analysis

To troubleshoot the Voice DSP Crash Dump File Analysis feature, use the **debug voice dsp crash-dump** command in privileged EXEC mode. This command is intended only for troubleshooting purposes because the volume of output generated by the software can result in severe performance degradation on the router.

SUMMARY STEPS

1. **enable**
2. **debug voice dsp crash-dump keepalive**
3. **undebug all**
4. **debug voice dsp crash detail**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
1.	**enable** Example: `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
2.	**debug voice dsp crash-dump keepalives** Example: `Router(config)#  no logging console`	Displays debugging information for the crash dump feature keepalives. • Confirms that a crash dump file has been written to the specified destination.
3.	**undebug all** Example: `Router#  un all`	Disables all the debug output on screen to stop the above output. • Alternately, you can use the **no debug all** command.
4.	**debug voice dsp crash detail** Example: `Router# debug voice dsp crash detail`	Displays debugging information for the crash dump feature details. • There is no debug output until there is one DSP crash. When the crash dump feature is turned on, the detailed debug messages are displayed.
5.	**exit** Example: `Router(config)#  exit`	Exits to privileged EXEC mode.

Verifying DSP Crash Dump File Analysis

To verify crash dump statistics, use the **show voice dsp crash-dump** command in privileged EXEC mode.

Configuration Examples for Voice DSP Crash Dump File Analysis

The following example shows that crash dump analysis is enabled:

```
Router(config)# voice dsp crash-dump destination 172.29.248.12
Router(config)# voice dsp crash-dump file-limit 10
voice dsp crash-dump destination tftp://172.29.248.12/tester/crash-152-t
voice dsp crash-dump file-limit 10
end
1w0d:%SYS-5-CONFIG_I:Configured from console by consoleoice dsp crash
```

Verifying Voice DSP Crash Dump File Analysis

The following example shows output information that verifies the status of the crash dump:

```
Router# show voice dsp crash-dump
Voice DSP Crash-dump status:
    Destination file url is
          tftp://172.29.248.12/tester/crash-152-t
    File limit is 10
    Last DSP dump file written was
          tftp://172.29.248.12/zongshan/crash/26-152-t2
    Next DSP dump file written will be
          tftp://172.29.248.12/tester/crash-152-t1
```

Troubleshooting Universal Port SPEs

A universal port card is a hardware card that processes digital signals for the Cisco AS5350 and Cisco AS5400 universal gateways. The service-processing element (SPE) works as a DSP for the universal port card.

This section provides troubleshooting information that apply to modems regardless of service type mode. It describes how to perform diagnostic tests on installed ports or SPEs, configure automatic recovery of ports on an SPE, and configure a scheduled recovery of SPEs.

Configure SPE Diagnostic Tests

You can perform three types of diagnostic tests on the SPE modem:

- SPE Startup Test
- SPE Auto-Test
- SPE Back-to-Back Test

SPE Startup Test

To perform diagnostic testing on all the installed SPE ports during the system's initial startup or rebooting process, use the **port modem startup-test** command in global configuration mode.

The results of the SPE port startup test are displayed in the **show port modem test** command output. SPE ports that pass the diagnostic test are marked as Pass, Fail, and Unkn. Ports that fail the diagnostic test are marked as Bad. These ports cannot be used for call connections. Depending on how many ports are installed, this diagnostic test may take from 5 to 10 minutes to complete. Perform additional testing on an inoperative SPE port by executing the **test port modem back-to-back** command. The **no port modem startup-test** command disables startup testing.

SPE Auto-Test

To perform diagnostic testing on all the installed SPE ports during the system's initial startup or rebooting process, or during service, use the **port modem autotest** command in global configuration mode.

The results of the SPE port auto-test are displayed in the **show port modem test** command's output. Ports that pass the diagnostic test are marked as *Idle*, *Busy*, *Downloading*, and *Reset,* and are put into service. Ports that fail the diagnostic test are marked as *Bad,* and are not put into service or tested again until they are no longer marked as *Bad*. If all the ports of an SPE are bad, the corresponding SPE is also marked bad. These ports cannot be used for call connections. Depending on how many ports are present and not marked *Bad*, this diagnostic test may take from 5 to 10 minutes to complete. You may perform additional testing on an inoperative port by executing the **test port modem back-to-back** command. The **no port modem autotest**command disables testing.

You may optionally configure the following commands:

- **port modem autotest minimum** *ports*-Define the minimum number of free ports available for autotest to begin.
- **port modem autotest time** *hh:mm {interval}*-Enable autotesting time and interval.A sample diagnostic autotest setting the time at 12:45 and at 8 hour intervals looks like the following:

```
AS5400(config)# port modem autotest time 12:45 8
AS5400(config)#
```

- **port modem autotest error** *threshold*-Define the maximum number of errors detected for autotest to begin.

SPE Back-to-Back Test

When an SPE port tests as *Bad*, perform additional testing by conducting a series of internal back-to-back connections and data transfers between two SPE ports. All port test connections occur inside the gateway. For example, if mobile users cannot dial into port 2/5 (the sixth port on the universal port card in the second chassis slot), attempt a back-to-back test with port 2/5 and a known-functioning port such as port 2/6.

Enter the following command in privileged EXEC mode (the prompt is displayed as AS5350# or AS5400#) to perform internal back-to-back port tests between two ports:

test port modem back-to-back *slot/port slot/port* {*num-packets*}-Perform internal back-to-back port tests between two ports, sending test packets of the specified size.

You might need to enable this command on several different combinations of ports to determine which one is not functioning properly. A pair of operable ports successfully connect and complete transmitting data in both directions. An operable port and an inoperable port do not successfully connect with each other.

A sample back-to-back test might look like the following:

```
AS5400# test port modem back-to-back 2/10 3/20
Repetitions (of 10-byte packets) [1]:
*Mar  02 12:13:51.743:%PM_MODEM_MAINT-5-B2BCONNECT:Modems (2/10) and (3/20) connected in back-to-
*Mar  02 12:13:52.783:%PM_MODEM_MAINT-5-B2BMODEMS:Modems (3/20) and (2/10) completed back-to-back
```

A port that has been confirmed to have problems can often be fixed using the **clear spe** command.

The results of the **test port modem back-to-back** command are displayed in the **show port modem test** command's output:

```
AS5400# show port modem test
 Date Time             Modem  Test               Reason            State Result
 3/02 12:00:57 PM       2/01  Back-To-Back     :STARTUP TEST      Idle  PASS
 3/02 12:00:57 PM       2/00  Back-To-Back     :STARTUP TEST      Idle  PASS
```

```
 3/02 12:00:58 PM       2/02  Back-To-Back     :STARTUP TEST      Idle  PASS
 3/02 12:00:58 PM       2/03  Back-To-Back     :STARTUP TEST      Idle  PASS
 3/02 12:00:58 PM       2/04  Back-To-Back     :STARTUP TEST      Idle  PASS
 3/02 12:00:58 PM       2/05  Back-To-Back     :STARTUP TEST      Idle  PASS
...
 3/02 12:01:14 PM       3/95  Back-To-Back     :STARTUP TEST      Idle  PASS
 3/02 12:01:14 PM       3/94  Back-To-Back     :STARTUP TEST      Idle  PASS
 3/02 12:01:15 PM       3/75  Back-To-Back     :STARTUP TEST      Idle  PASS
 3/02 12:01:15 PM       3/74  Back-To-Back     :STARTUP TEST      Idle  PASS
 3/02 12:13:52 PM       3/20  Back-To-Back     :USER INITIATED    Idle  PASS
 3/02 12:13:52 PM       2/10  Back-To-Back     :USER INITIATED    Idle  PASS
...
 3/02 12:44:00 PM      3/102  No Test (Time)   :MIN IDLE MODEMS   Idle  NOTST
 3/02 12:44:00 PM      3/103  No Test (Time)   :MIN IDLE MODEMS   Idle  NOTST
 3/02 12:44:00 PM      3/104  No Test (Time)   :MIN IDLE MODEMS   Idle  NOTST
 3/02 12:44:00 PM      3/105  No Test (Time)   :MIN IDLE MODEMS   Idle  NOTST
 3/02 12:44:00 PM      3/106  No Test (Time)   :MIN IDLE MODEMS   Idle  NOTST
 3/02 12:44:00 PM      3/107  No Test (Time)   :MIN IDLE MODEMS   Idle  NOTST
 3/02 12:44:21 PM       2/73  Back-To-Back     :TIME INTERVAL     Idle  PASS
 3/02 12:44:21 PM       2/72  Back-To-Back     :TIME INTERVAL     Idle  PASS
 3/02 12:44:21 PM       2/33  Back-To-Back     :TIME INTERVAL     Idle  PASS
 3/02 12:44:21 PM       2/32  Back-To-Back     :TIME INTERVAL     Idle  PASS
 3/02 12:44:21 PM       3/37  Back-To-Back     :TIME INTERVAL     Idle  PASS
```

Note: The *Reason* column indicates why the test was started. The *TIME INTERVAL* is one of the triggers under autotest; the other is the error threshold.

SPE Disconnect Reason Codes

This section describes how to interpret the call disconnect reason codes reported by Cisco universal port card SPEs. Whenever a call using the universal port SPEs is cleared or disconnected, the SPE records the reason for the disconnect. This disconnect reason code can be used to determine whether the disconnect was normal or an error occurred. This reason code can be used to track down possible sources of failure. Modems can be disconnected due to a variety of factors such as client disconnects, telco errors, and call drops at the network access server (NAS). A "good" disconnect reason is that the DTE (client modem or NAS) at one end or the other wanted to terminate the call. Such "normal" disconnects indicate that the disconnect was not a result of modem or transmission level errors.

Note: The disconnect reason is managed in a first-come-first-serve fashion. This means that the first disconnect reason generated is the only disconnect reason recorded. If the modem and the NAS attempt to terminate the session simultaneously and the modem happens to save the disconnect reason before the LINK_TERMINATE message from the NAS is processed, then the NAS disconnect reason is ignored.

Determining the disconnect Reason

When evaluating whether you are experiencing "good" or "bad" disconnects, it is important to obtain the history of disconnects that a particular port has experienced. In most environments, the disconnect reason is obtained using modem call records or call tracker syslog messages. This disconnect code can then be interpreted using the table provided in this document. Use the following commands to determine the disconnect reason:

- The **show spe modem disconnect-reason** command does not display the disconnect reason code as a hexadecimal value. However, it does indicate the disconnect reason as a name. The name and class of the disconnect reason can be found inTable: CLASS OTHER Code Summary Table and Table: CLASS DSP Reason Codes respectively.
- The **show port modem log** command displays the disconnect reason code as a hexadecimal value. Refer to Table: Disconnect Reason Code Hexadecimal Values for the hexadecimal values.

Table: Disconnect Reason Code Hexadecimal Values

0x0..		0x1..			0x2..			0x3..	0x4..	0x5..
-	0x010	0x100	0x1F00	-	-	0x210	0x220	0x3xx	-	-
0x001	0x011	0x101	0x1F01	-	0x201	0x211	0x221	-	-	0x501
0x002	0x012	0x102	0x1F02	-	0x202	0x212	0x222	-	-	0x502
0x003	-	0x103	0x1F03	-	0x203	-	-	-	0x403	0x503
0x004	-	0x104	0x1F04	-	0x204	-	0x224	-	0x404	0x504
0x005	-	0x105	0x1F05	-	0x205	-	0x225	-	-	0x505
0x006	-	0x106	0x1F06	-	0x206	-	-	-	-	0x506
0x007	-	0x107	0x1F07	-	-	-	-	-	-	0x507
0x008	-	0x108	0x1F08	-	-	-	-	-	0x408	-
0x009	-	0x109	-	-	-	-	-	-	-	-
0x00C	-	-	-	-	-	-	-	-	-	-
0x00D	-	-	-	-	-	-	-	-	-	-
0x00E	-	-	-	-	-	-	-	-	-	-
0x00F	-	-	-	0x1FFF	-	-	-	-	-	-

Using the show port modem log command

Use the **show port modem log** *slot/port* command to obtain the disconnect cause code (in Hex) for a particular call on a specific port. This disconnect code is identical to the cause code obtained from modem call-record and call-tracker syslog outputs. An example is shown:

```
*Jan  1 00:53:56.867: Modem State event: State: Terminate
*Jan  1 00:53:56.879: Modem End Connect event:
  Call Timer                              :   195  secs
  Disconnect Reason Info                  :   0x220
      Type (=0  ):
     Class (=2  ):  EC condition - locally detected
    Reason (=32 ):  received DISC frame -- normal LAPM termination
```

From the example above, note that the disconnect code is **0x220**.

Using the show spe modem disconnect-reason Command

Use the **show spe modem disconnect-reason** {**summary** | *slot* | *slot/spe*} command to determine the distribution of disconnect reasons that the particular port has experienced. A sample summary output of all the ports is shown below:

```
Router# show spe modem disconnect-reason summary
     ===CLASS OTHER====  =====CLASS DSP====  ===CLASS EC LCL===  ==CLASS EC FRMR===
     Software Rst     0  No Carrier     341  No LR            0  Frmr Bad Cmd     0
     EC Termntd       0  No ABT dtctd     0  LR Param1        0  Frmr Data        0
     Bad MNP5 Rx      0  Trainup flr    328  LR Incmpt        0  Frmr Length      0
     Bad V42B       110  Retrain Lt       0  Retrns Lt      226  Frmr Bad NR      0
     Bad COP stat     0  ABT end flr      0  Inactivity       0
     ATH              0                     Protocol Err     1  ===CLASS EC LD====
     Aborted          0  ====CLASS HOST====  Fallbck Term    74  LD No LR         0
     Connect Tout   198  Hst NonSpec      0  No XID          67  LD LR Param1     0
     Reset DSP        0  HST Busy         0  XID Incmpt       0  LD LR Incmpt     0
                         HST No answr     0  Disc         21448  LD Retrns Lt     0
     ===CLASS EC Cmd===  HST DTR       3615  DM               5  LD Inactivty     0
     Bad Cmd          0  HST ATH          0  Bad NR           0  LD Protocol      0
                         HST NoDialTn     0  SABME Online     0  LD User          0
```

```
=====N O N E======  HST No Carr   5276  XID Online       0
None            39  HST Ack          0  LR Online        0  TOTAL  31728
HST NoDialTn     0  SABME Online     0  LD User          0=====N O N E======
HST No Carr   5276  XID Online       0  None            39  HST Ack          0
LR Online        0  TOTAL        31728
```

From the example above, let us say that we are interested in the disconnect category Disc within **CLASS EC LCL**. To determine what the disconnect reason Disc means, go to the entry corresponding to the class (CLASS EC LCL) and the disconnect reason name (Disc) which shows a hex code of 0x220 and is a normal disconnect. The codes for the classes are shown in the following tables:

- Table: CLASS OTHER Code Summary Table
- Table: CLASS DSP Reason Codes
- Table: CLASS EC LCL: EC Condition, Locally Detected Reason Code Table
- Table: CLASS EC Cmd: EC Detected Bad Command Code Reason Code Table
- CLASS EC FRMR: EC Detected FRMR From Peer Reason Code Table
- Table: CLASS EC LD: Error Correction (EC) Detected Link Disconnect (LD) From Peer Reason Code Table
- Table: CLASS HOST: Requested by Host Reason Code Table

Reason Code Summary Tables

The following tables contain the detailed reason codes for universal port disconnects.

Table: CLASS OTHER Code Summary Table

Disconnect Reason Type	Disconnect Reason: Name	Disconnect Reason Code (Hex)	Description
2	Software Rst	0x001	Cisco IOS software disconnected the call for some indeterminate reason (SOFTWARE_RESET).
2	EC Termntd	0x002	Error Correction (EC) layer termination
2	Bad MNP5 Rx	0x003	The Microcom Network Protocol 5 (MNP5) decompression task received an illegal token in the data stream. There is probably a logic error in the implementation of compression, decompression or error correction by the modem or partner. Can also be caused by a transient line or RAM memory error.
2	Bad V42B	0x004	The V.42bis or V.44 decompression task received an illegal token in the data stream. There is probably a logic error in either the modem's or partner's implementation of compression, decompression or error correction. Can also be caused by a transient line or RAM memory error.
2	Bad COP stat	0x005	Reserved
6,7	ATH	0x006	ATH command detected by local modem. The ATH (Hangup) AT command is detected by the local modem (universal port card). For example, following a dialout from Cisco IOS, the DTE interface clears the call by transmitting an inband ATH AT command after the call is connected.
3	Aborted	0x007	AT mode any-key abort of dial command The AT dial command was aborted by the any-key abort command. For example, the

			host modem originates a call. During connection establishment, pressing any-key causes the AT dial command to be aborted.
3	Connect Tout	0x008	The call took too long to complete the connection. Notice that the S7 timer (wait for carrier after dial) expired for this disconnect. The causes include: • Difficulty negotiating a Layer I standard • Layer I and Layer II establishment taking too long. For example, error correction negotiation takes an extended amount of time because of bit-errors introduced when the client modem receiver tries to connect at a rate it can't sustain. This disconnect could also happen if the answer modem heard no tone from the channel because, for example, the originator was not a modem.
2	Reset DSP	0x009	The DSP was reset (command/internal/spontaneous). The DSP within the host modem was reset by the Control Processor (CP) or Signal Processor (SP). The CP resets the DSP if mail messages from the CP to SP are not being acknowledged. The SP resets itself if it gets an internal inconsistency error.
4,6	-	0x00C	V.42bis or V.44 codeword size exceeded negotiated maximum.
4,6	-	0x00D	V.42bis or V.44 received codeword equal to next empty dictionary entry.
4,6	-	0x00E	V.42bis or V.44 received codeword greater than the next empty dictionary entry.
4,6	-	0x00F	V.42bis or V.44 received reserved command code.
4,6	-	0x010	V.42bis or V.44 ordinal size exceeded eight.
4,6	-	0x011	V.42bis or V.44 negotiation error.
4,6	-	0x012	V.42bis or V.44 compression error.

Table: CLASS DSP Reason Codes

Disconnect Reason Type	Disconnect Reason: Name	Disconnect Reason Code (Hex)	Description
-	-	0x1xx	DSP conditions reported by SPE
4,5	No Carrier	0x100	The SPE carrier signal is lost. The universal port card detected a client modem carrier drop. The universal port SPE stopped hearing carrier for a period greater than the value specified in Register S10 (hang-up delay after carrier loss). This could mean that the talk path went away or that the client stopped transmitting. If a layer II protocol (V.42 and/or V.42bis) is in effect, it is abnormal to see such a

			disconnect. Common causes are users hanging up the call before a connection takes place can occur because of incidental dialing, aborted starts, and client applications timing out when calls take too long to connect due to multiple retrains during Layer 1 negotiation. The condition can also occur during normal data mode when the client abruptly drops the carrier. This can occur if the link is abruptly dropped (network error), or power is shut off to the client modem disconnecting the call. This can also occur with less sophisticated client modems that do not implement the Layer I and/or Layer II clear-down protocols on a DTR drop. For a large number of client modems, this is considered a normal disconnection.
3	No ABT dtctd	0x101	No answer-back tone detected, caller is probably not a modem
3	Trainup flrv	0x102	Call failure while modem training up due to incompatible modulation or bad line. This may be indicative of attempts to negotiate an unsupported modulation such as a legacy Rockwell proprietary modulation (K56Plus, V.FC, and so on). Other possible causes are DSP failures to train up due to severe line impairments, impulse noises, interrupting training, incompatible modulation parameters, and perhaps the inability to properly select a Layer I standard.
4,5	Retrain Lt	0x103	Too many consecutive retrains or speed-shifts. The retrain limit is specified with Register S40. During the progress of a call, too many retrains occurred and rendered the call ineffective-the data rate would be so poor as to be useless. Sometimes the client modem does not complete the clear-down protocol, for example, when the Telco tore down the call in the middle of the connection; NextPort (NP) attempts to recover the call by issuing retrains. Once the retrain limit is reached, NP drops the call and report this disconnect reason.
3	ABT end flr	0x104	Problem detecting end of Answer-Back Tone(ABT). Negotiation failure or excessive noise during V.34 training. Host modems answer and send out V.8bis and modulated 2100Hz answer-back tones (ABTs) with phase reversals but encounter excessive noise during the trainup sequence. Look for errors on the path from the calling modem to the answering modem in either one or both directions. Similar behavior occurs when there is latency in the Public Switched Telephone Network (PSTN)that exceeds one second for dial up and causes modems to be unable to train up the echo cancellers. Other possible causes are: • TX power levels are incorrect and the tones are then not handled by the remote side.

			• Excessive noise in Phase III and IV during V.34 training. • Operator error. • Network interference during V.34 training (someone picks up the extension).
3	-	0x105	SS7/COT (continuity test) operation completed successfully.
3	-	0x106	SS7/COT (continuity test) operation failed: T8/T24 timeout waiting for tone on.
3	-	0x107	SS7/COT (continuity test) operation failed: T8/T24 timeout waiting for tone off.
4	-	0x108	Modem on hold (MOH) cleardown by the universal port card. V.92 specifies that the cleardown reason can be: • Cleardown due to incoming call • Cleardown due to outgoing call • Cleardown due to other reason
4	-	0x109	MOH timeout value reached. This value can be adjusted using Register S62 (V.92 maximum MOH time).

Table: CLASS EC LCL: EC Condition, Locally Detected Reason Code Table

Disconnect Reason Type	Disconnect Reason: Name	Disconnect Reason Code (Hex)	Description
-	-	0x2xx	Local error correction (EC) conditions.
3	No LR	0x201	During negotiation a link request (LR) frame was not received. Peer may not support MNP.
3	LR Param1	0x202	The received MNP LR frame had a bad/unexpected PARAM1. For more information on PARAM1 refer to the V.42 specification.
3	LR Incmpt	0x203	The received MNP LR frame is incompatible with the host modem's settings for EC.
4,5	Retrns Lt	0x204	Too many consecutive retransmissions in EC. This disconnect reason can be caused by noise on the line that spurs retransmissions. For instance, the host modem transmits data to the client modem, but noise on the line causes the data to be received incorrectly (or not at all) by the client side. The client modem could also have disconnected without the host modem realizing this. So the host modem continuously retransmits, without knowing that the client modem is no longer present. Sometimes, when the call connects in LAPM or MNP, the universal port card is unable to transmit a frame to the client modem. The client modem fails to acknowledge the universal port card's initial transmission, then fails to respond to Register S19 (error correction retransmission limit) polls (the default is 12), so NP disconnects the call. One cause could be that the carrier in the transmit path degraded substantially while the client failed to downshift. Another cause could be a problem with the client's EC

			engine (as happens on a Winmodem system if Windows stops responding).
6,7	Inactivity	0x205	Inactivity timeout, MNP Link Disconnect (LD) sent. The host modem sends the client modem a LD frame, indicating that an inactivity timeout has occurred.
4,5	Protocol Err	0x206	EC protocol error. This is a general catch-all protocol error. It indicates that a LAPM or MNP EC protocol error has occurred.
3	Fallbck Term	0x210	No EC fallback protocol available. Error correction negotiation has not been successful. The call is terminated because there is no error correction fallback protocol available. S-register S25 (link protocol fallback) determines the available fallback protocol. The options are asynchronous framing, synchronous framing, or disconnect (hang up).
3	No XID	0x211	Never received eXchange IDentification (XID) frame during negotiation. Peer may not support MNP.
3	XID Incmpt	0x212	The received XID frame is incompatible with local settings. The client modem may not support LAPM within V.42.
3,4,5	Disc	0x220	Received Disconnect (DISC) frame. This is the normal LAP-M disconnect. The call terminated normally with a proper clear down from the client side. For example, a V.42 disconnect packet was sent from the client modem to the host modem. The client modem dropped DTR and cleanly negotiated a clear-down protocol.
3,4,5	DM	0x221	Received DM frame. Peer might be disconnecting. The client modem indicates that it is disconnecting. During call setup, this reason indicates that the client modem is giving up on negotiating error correction.
4,5	Bad NR	0x222	Bad receive sequence number or ACK number was received. An MNP LD or LAP-M FRMR is sent. The host modem received a LAPM or MNP error correction frame with a bad sequence number or acknowledgment number. An LD or Frame Reject (FRMR) frame is sent to the client modem, indicating that the host modem is disconnecting.
4,5	SABME Online	0x224	Received MNP XID frame in steady-state. This is interpreted as a LAPM error correction protocol error in steady state. It means that the client modem may have reset due to receiving a FRMR.
4,5	XID Online	0x225	Received MNP LR frame while in steady-state. This is interpreted as an MNP error correction protocol error in steady state. It means that the client modem has reset.

Table: CLASS EC Cmd: EC Detected Bad Command Code Reason Code Table

!Disconnect Reason Type	!Disconnect Reason: Name	!Disconnect Reason Code (Hex)	!Description
4,5	Bad Cmd	0x3xx	EC detected bad command code. The received unknown command is in the last 2 digits. An MNP LD or LAP-M FRMR frame is sent in response.

Table: CLASS EC FRMR: EC Detected FRMR From Peer Reason Code Table

Disconnect Reason Type	Disconnect Reason: Name	Disconnect Reason Code (Hex)	Description
4,5	-	0x4xx	EC conditions indicated by client in LAP-M FRMR frame. The bit-mapped reason is in the last two digits.
4,5	Frmr Bad Cmd	0x401	LAPM: peer reports bad command. The host modem received a FRMR frame from the client modem. The received FRMR frame indicates that the client modem received an error correction frame from the host modem that contained a bad command.
4,5	Frmr Data	0x403	LAPM: peer reports that data field is not permitted or is incorrect length (U frames). The host modem received a FRMR frame from the client modem. The received FRMR frame indicates that the client modem received an error correction frame from the host modem that contained a data field that is not permitted or contained a data field with an incorrect length (that is, U frame).
4,5	Frmr Length	0x404	LAPM: peer reports data field length is greater than N401 (the maximum information field length specified in V.42), but has good Frame Check Sequence (FCS). The modem received a FRMR frame from the client modem. The received FRMR frame indicates that the client modem received an error correction frame from the modem that contained a data field length that is greater than the maximum number of octets that can be carried in the information field (N401) of an I frame, an SREJ frame, an XID frame, a UI frame, or a TEST frame. The frame check sequence is good.
4,5	Frmr Bad NR	0x408	LAPM: peer reports bad receive sequence number or N(R). The host modem received a FRMR frame from the client modem. The received FRMR frame indicates that the client modem received an error correction frame from the host modem that contained a bad receive sequence number.

Table: CLASS EC LD: Error Correction (EC) Detected Link Disconnect (LD) From Peer Reason Code Table

Disconnect Reason Type	Disconnect Reason: Name	Disconnect Reason Code (Hex)	Description

4,5	-	0x5xx	EC conditions indicated by client in MNP link disconnect (LD) frame. Reason field is in the last 2 digits.
3	LD No LR	0x501	MNP: peer never received LR frame. The host modem received a LD frame from the client modem. The received LD frame indicates that the client modem never received a link request from the host modem.
3	LD LR Param1	0x502	MNP: peer reports link request (LR) frame has bad parameter #1 The host modem received an LD frame from the client modem. The received LD frame indicates that the client modem received a link request frame from the host modem that contained an unexpected PARAM1. For more information on PARAM1 refer to the V.42 specification.
3	LD LR Incmpt	0x503	MNP: peer reports LR frame is incompatible with its configuration The host modem received an LD frame from the client modem. The received LD frame indicates that the client modem received an LR frame from the host modem that is incompatible with the configuration of the client modem.
4,5	LD Retrns Lt	0x504	MNP: peer reports too many consecutive EC retransmissions The host modem received a LD frame from the client modem. The received LD frame indicates that the client modem received too many consecutive retransmissions.
4,5	LD Inactivty	0x505	MNP: peer reports inactivity timer expired The host modem received a Link Disconnect (LD) frame from the client modem. The received LD frame indicates that the client modem's host (DTE) has not passed data to the client modem within a period of time.
3	LD Protocol	0x506	MNP: peer reports error The host modem received an LD frame from the client modem. The received LD frame indicates that the client modem received a MNP protocol error.
3	LD User	0x507	Normal MNP disconnect The host modem received a LD frame from the client modem. The received LD frame indicates a normal MNP termination.

Table: CLASS HOST: Requested by Host Reason Code Table

Disconnect Reason Type	Disconnect Reason: Name	Disconnect Reason Code (Hex)	Description
6,7	-	0x1Fxx	Host initiated disconnect. Value is a sum of 0x1F00 and SessionStopCommand value. This is the other host terminate reason. The host reason is indicated in the low-order bytes "xx".
3,6,7		0x1F00	

Table: CLASS EC LD: Error Correction (EC) Detected Link Disconnect (LD) From Peer Reason Code Table

	HST NonSpec		Non-specific host-initiated disconnect. Value is a sum of 0x1F00 and SessionStopCommand value. This is the catch all Cisco IOS-initiated disconnect reason. It is used for all non-standard disconnects. For example, this could be a result of modem management software deciding to terminate the call. One possible explanation is a higher-level authentication failure RADIUS, TACACS, or another application issuing a DTR drop to the host modem. This type of disconnect does not count towards CSR when the host modem is in data mode.
3	HST Busy	0x1F01	Dialed number was busy. Disconnection has occurred because the host is indicating that the dialed number is busy.
3	HST No answr	0x1F02	Dialed number did not answer. Disconnection has occurred because the host is indicating that the dialed number didn't answer.
3,6,7	HST DTR	0x1F03	Virtual DTR dropped. This status is reflected by the I/O port redirector that is currently using the modem. Disconnection has occurred because the host dropped the virtual DTR line. This generic disconnect cause is initiated by the Cisco IOS software. Example causes are idle timeout, PPP LCP TERMREQ received, authentication failure, Telnet hangup, and so on. To determine the reason for the hang up, examine the Radius disconnect reason from the modem call-record terse command or from Authentication, Authorization, and Accounting (AAA).
6,7	HST ATH	0x1F04	ATH (hangup) command was detected by local host.
3	HST NoDialTn	0x1F05	No access to telco network. Disconnection has occurred because the host could not access the network.
3,4,5	HST NoCarr	0x1F06	Network indicated disconnect. This is a client-side triggered disconnect that is not a graceful call termination. It can occur during call set-up. A common cause is when users of Windows 95 or Windows 98 Dial Up Networking (DUN) cancel the call before the call reaches steady state. Another common reason is any client-instigated DTR drop before steady state. During data mode, this is also a client side triggered disconnection that is not a graceful call termination. One very common cause is authentication failures.
3	-	0x1F07	NAS terminated SS7/continuity test (COT) operation. Disconnection has occurred because the NAS has terminated the SS7/COT operation.
3	-	0x1F08	The SS7/COT operation was terminated by the router because of a T8/T24 timeout.
-	-	0x1FFF	Unsolicited TERMINATING. The host sends this disconnect reason when it receives a unsolicited terminating message.

Table: Disconnect Reason Types

!Disconnect Type	!Description
0	(unused)
1 - 0x2...	(unused)
2 - 0x4...	Other situations
3 - 0x6...	Condition occurred during call setup
4 - 0x8...	In data mode. Rx (line to host) data flushing OK
5 - 0xA...	In data mode. Rx (line to host) data flushing not OK (at present, applications should not be concerned about the "not OK")
6 - 0xC...	In data mode. Tx (host to line) data flushing OK
7 - 0xE...	In data mode. Tx (host to line) data flushing not OK (at present, applications should not be concerned about the "not OK")

For more information about troubleshooting SPEs, refer to Interpreting NextPort Disconnect Reason Codes, document ID 9502.

DSP Troubleshooting Links

DSP troubleshooting procedures vary from platform to platform. To troubleshoot DSPs on your Cisco product, see the following links:

- For troubleshooting the DSP on NM-HDV for Cisco 2600 series, Cisco 3600 series, and VG200 series routers, refer to Troubleshooting the DSP on NM-HDV for Cisco 2600/3600/VG200 Series Routers, document ID 19066.
- For troubleshooting the VTSP-3-DSP_TIMEOUT error on AS5300 platforms, refer to Troubleshooting VTSP-3-DSP_TIMEOUT Error on Cisco AS5300 Access Server Platforms, document ID 18680.
- If you have problems with unrecognized voice cards on Cisco 1750, Cisco 1751, and Cisco 1760 routers, it could be a problem with the packet voice data module (PVDM), which houses the DSPs. Refer to Troubleshooting Unrecognized Voice Interface Cards on Cisco 1750, 1751, and 1760 Routers, document ID 5711.