debug modem

To observe modem line activity on an access server, use the **debug modem** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem

no debug modem

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debug modem** command:

```
Router# debug modem

15:25:51: TTY4: DSR came up
15:25:51: tty4: Modem: IDLE->READY
15:25:51: TTY4: Autoselect started
15:27:51: TTY4: Autoselect failed
15:27:51: TTY4: Line reset
15:27:51: TTY4: Modem: READY->HANGUP
15:27:52: TTY4: dropping DTR, hanging up
15:27:52: tty4: Modem: HANGUP->IDLE
15:27:57: TTY4: restoring DTR
15:27:58: TTY4: DSR came up
```

The output shows when the modem line changes state.

debug modem csm

To debug the Call Switching Module (CSM), used to connect calls on the modem, use the **debug modem csm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem csm [*slot/port* | **group** *group-number*]

no debug modem csm [*slot/port* | **group** *group-number*]

Syntax Description

slot/port	(Optional) The slot and modem port number.
group *group-number*	(Optional) The modem group.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debug modem csm** command to troubleshoot call switching problems. With this command, you can trace the complete sequence of switching incoming and outgoing calls.

Examples

The following is sample output from the **debug modem csm** command. In this example, a call enters the modem (incoming) on slot 1, port 0:

```
Router(config)# service timestamps debug uptime

Router(config)# end

Router# debug modem csm

00:04:09: ccpri_ratetoteup bear rate is 10
00:04:09: CSM_MODEM_ALLOCATE: slot 1 and port 0 is allocated.
00:04:09: MODEM_REPORT(0001): DEV_INCALL at slot 1 and port 0
00:04:09: CSM_PROC_IDLE: CSM_EVENT_ISDN_CALL at slot 1, port 0
00:04:11: CSM_RING_INDICATION_PROC: RI is on
00:04:13: CSM_RING_INDICATION_PROC: RI is off
00:04:15: CSM_PROC_IC1_RING: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0
00:04:15: MODEM_REPORT(0001): DEV_CONNECTED at slot 1 and port 0
00:04:15: CSM_PROC_IC2_WAIT_FOR_CARRIER: CSM_EVENT_ISDN_CONNECTED at slot 1, port 0
```

The following is sample output from the **debug modem csm** command when call is dialed from the modem into the network (outgoing) from slot 1, port 2:

```
Router# debug modem csm

atdt16665202
00:11:21: CSM_PROC_IDLE: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 2
00:11:21: T1_MAIL_FROM_NEAT: DC_READY_RSP: mid = 1, slot = 0, unit = 0
00:11:21: CSM_PROC_OC1_REQUEST_DIGIT: CSM_EVENT_DIGIT_COLLECT_READY at slot 1, port 2
00:11:24: T1_MAIL_FROM_NEAT: DC_FIRST_DIGIT_RSP: mid = 1, slot = 0, unit = 0
00:11:24: CSM_PROC_OC2_COLLECT_1ST_DIGIT: CSM_EVENT_GET_1ST_DIGIT at slot 1, port 2
00:11:27: T1_MAIL_FROM_NEAT: DC_ALL_DIGIT_RSP: mid = 1, slot = 0, unit = 0
00:11:27: CSM_PROC_OC3_COLLECT_ALL_DIGIT: CSM_EVENT_GET_ALL_DIGITS (16665202) at slot 1,
port 2
00:11:27: ccpri_ratetoteup bear rate is 10
```

```
00:11:27: MODEM_REPORT(A000): DEV_CALL_PROC at slot 1 and port 2
00:11:27: CSM_PROC_OC4_DIALING: CSM_EVENT_ISDN_BCHAN_ASSIGNED at slot 1, port 2
00:11:31: MODEM_REPORT(A000): DEV_CONNECTED at slot 1 and port 2
00:11:31: CSM_PROC_OC5_WAIT_FOR_CARRIER: CSM_EVENT_ISDN_CONNECTED at slot 1, port 2
CONNECT 19200/REL - MNP
```

The following is sample output from the **debug modem csm** command for an incoming call:

```
Router# debug modem csm

Router#1.19.36.7 2001
Trying 1.19.36.7, 2001 ... Open
atdt111222333444555666
*Apr  7 12:39:42.475: Mica Modem(1/0): Rcvd Dial String(111222333444555666)
*Apr  7 12:39:42.475: CSM_PROC_IDLE: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0
*Apr  7 12:39:42.479: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_CHANNEL_LOCK at slot 1 and
port 0
*Apr  7 12:39:42.479: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_BCHAN_ASSIGNED at slot 1, port
0
*Apr  7 12:39:42.479: Mica Modem(1/0): Configure(0x1)
*Apr  7 12:39:42.479: Mica Modem(1/0): Configure(0x5)
*Apr  7 12:39:42.479: Mica Modem(1/0): Call Setup
*Apr  7 12:39:42.479: neat msg at slot 0: (1/0): Tx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:42.491: neat msg at slot 0: (0/0): Rx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:42.531: VDEV_ALLOCATE: slot 1 and port 3 is allocated.
*Apr  7 12:39:42.531: CSM_RX_CAS_EVENT_FROM_NEAT:(0004):  EVENT_CALL_DIAL_IN at slot 1 and
port 3
*Apr  7 12:39:42.531: CSM_PROC_IDLE: CSM_EVENT_DSX0_CALL at slot 1, port 3
*Apr  7 12:39:42.531: Mica Modem(1/3): Configure(0x0)
*Apr  7 12:39:42.531: Mica Modem(1/3): Configure(0x5)
*Apr  7 12:39:42.531: Mica Modem(1/3): Call Setup
*Apr  7 12:39:42.595: Mica Modem(1/0): State Transition to Call Setup
*Apr  7 12:39:42.655: Mica Modem(1/3): State Transition to Call Setup
*Apr  7 12:39:42.655: Mica Modem(1/3): Went offhook
*Apr  7 12:39:42.655: CSM_PROC_IC1_RING: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 3
*Apr  7 12:39:42.671: neat msg at slot 0: (0/0): Tx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:42.691: neat msg at slot 0: (1/0): Rx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:42.731: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_START_TX_TONE at slot 1
and port 0
*Apr  7 12:39:42.731: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_START_TX_TONE at slot 1, port 0
*Apr  7 12:39:42.731: Mica Modem(1/0): Generate digits:called_party_num= len=1
*Apr  7 12:39:42.835: Mica Modem(1/3): Rcvd Digit detected(#)
*Apr  7 12:39:42.835: CSM_PROC_IC2_COLLECT_ADDR_INFO: CSM_EVENT_KP_DIGIT_COLLECTED (DNIS=,
ANI=) at slot 1, port 3
*Apr  7 12:39:42.855: neat msg at slot 0: (0/0): Tx LOOP_OPEN (ABCD=0101)
*Apr  7 12:39:42.871: neat msg at slot 0: (1/0): Rx LOOP_OPEN (ABCD=0101)
*Apr  7 12:39:42.899: Mica Modem(1/0): Rcvd Digits Generated
*Apr  7 12:39:42.911: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_END_TX_TONE at slot 1 and
port 0
*Apr  7 12:39:42.911: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_END_TX_TONE at slot 1, port 0
*Apr  7 12:39:42.911: Mica Modem(1/0): Generate digits:called_party_num=A len=1
*Apr  7 12:39:43.019: Mica Modem(1/0): Rcvd Digits Generated
*Apr  7 12:39:43.019: CSM_PROC_OC4_DIALING: CSM_EVENT_TONE_GENERATED at slot 1, port 0
*Apr  7 12:39:43.019: Mica Modem(1/3): Rcvd Digit detected(A)
*Apr  7 12:39:43.335: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_START_TX_TONE at slot 1
and port 0
*Apr  7 12:39:43.335: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_START_TX_TONE at slot 1, port 0
*Apr  7 12:39:43.335: Mica Modem(1/0): Generate digits:called_party_num=111222333444555666
len=19
*Apr  7 12:39:43.439: Mica Modem(1/3): Rcvd Digit detected(1)
*Apr  7 12:39:43.559: Mica Modem(1/3): Rcvd Digit detected(1)
*Apr  7 12:39:43.619: Mica Modem(1/3): Rcvd Digit detected(1)
*Apr  7 12:39:43.743: Mica Modem(1/3): Rcvd Digit detected(2)
*Apr  7 12:39:43.859: Mica Modem(1/3): Rcvd Digit detected(2)
```

```
*Apr  7 12:39:43.919: Mica Modem(1/3): Rcvd Digit detected(2)
*Apr  7 12:39:44.043: Mica Modem(1/3): Rcvd Digit detected(3)
*Apr  7 12:39:44.163: Mica Modem(1/3): Rcvd Digit detected(3)
*Apr  7 12:39:44.223: Mica Modem(1/3): Rcvd Digit detected(3)
*Apr  7 12:39:44.339: Mica Modem(1/3): Rcvd Digit detected(4)
*Apr  7 12:39:44.459: Mica Modem(1/3): Rcvd Digit detected(4)
*Apr  7 12:39:44.523: Mica Modem(1/3): Rcvd Digit detected(4)
*Apr  7 12:39:44.639: Mica Modem(1/3): Rcvd Digit detected(5)
*Apr  7 12:39:44.763: Mica Modem(1/3): Rcvd Digit detected(5)
*Apr  7 12:39:44.883: Mica Modem(1/3): Rcvd Digit detected(5)
*Apr  7 12:39:44.943: Mica Modem(1/3): Rcvd Digit detected(6)
*Apr  7 12:39:45.063: Mica Modem(1/3): Rcvd Digit detected(6)
*Apr  7 12:39:45.183: Mica Modem(1/3): Rcvd Digit detected(6)
*Apr  7 12:39:45.243: Mica Modem(1/3): Rcvd Digit detected(B)
*Apr  7 12:39:45.243: CSM_PROC_IC2_COLLECT_ADDR_INFO: CSM_EVENT_DNIS_COLLECTED
(DNIS=111222333444555666, ANI=) at slot 1, port 3
*Apr  7 12:39:45.363: Mica Modem(1/0): Rcvd Digits Generated
*Apr  7 12:39:45.891: neat msg at slot 0: (0/0): Tx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:45.907: neat msg at slot 0: (1/0): Rx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:46.115: neat msg at slot 0: (0/0): Tx LOOP_OPEN (ABCD=0101)
*Apr  7 12:39:46.131: neat msg at slot 0: (1/0): Rx LOOP_OPEN (ABCD=0101)
*Apr  7 12:39:46.175: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_START_TX_TONE at slot 1
and port 0
*Apr  7 12:39:46.175: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_START_TX_TONE at slot 1, port 0
*Apr  7 12:39:46.175: Mica Modem(1/0): Generate digits:called_party_num= len=3
*Apr  7 12:39:46.267: Mica Modem(1/3): Rcvd Digit detected(#)
*Apr  7 12:39:46.387: Mica Modem(1/3): Rcvd Digit detected(A)
*Apr  7 12:39:46.447: Mica Modem(1/3): Rcvd Digit detected(B)
*Apr  7 12:39:46.447: CSM_PROC_IC2_COLLECT_ADDR_INFO: CSM_EVENT_ADDR_INFO_COLLECTED
(DNIS=111222333444555666, ANI=) at slot 1, port 3
*Apr  7 12:39:46.507: Mica Modem(1/0): Rcvd Digits Generated
*Apr  7 12:39:46.507: CSM_PROC_OC4_DIALING: CSM_EVENT_ADDR_INFO_COLLECTED at slot 1, port
0
*Apr  7 12:39:47.127: CSM_RX_CAS_EVENT_FROM_NEAT:(0004):  EVENT_CHANNEL_CONNECTED at slot
1 and port 3
*Apr  7 12:39:47.127: CSM_PROC_IC4_WAIT_FOR_CARRIER: CSM_EVENT_DSX0_CONNECTED at slot 1,
port 3
*Apr  7 12:39:47.127: Mica Modem(1/3): Link Initiate
*Apr  7 12:39:47.131: neat msg at slot 0: (0/0): Tx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:47.147: neat msg at slot 0: (1/0): Rx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:47.191: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_CHANNEL_CONNECTED at slot
1 and port 0
*Apr  7 12:39:47.191: CSM_PROC_OC5_WAIT_FOR_CARRIER: CSM_EVENT_DSX0_CONNECTED at slot 1,
port 0
*Apr  7 12:39:47.191: Mica Modem(1/0): Link Initiate
*Apr  7 12:39:47.227: Mica Modem(1/3): State Transition to Connect
*Apr  7 12:39:47.287: Mica Modem(1/0): State Transition to Connect
*Apr  7 12:39:49.103: Mica Modem(1/0): State Transition to Link
*Apr  7 12:39:52.103: Mica Modem(1/3): State Transition to Link
*Apr  7 12:40:00.927: Mica Modem(1/3): State Transition to Trainup
*Apr  7 12:40:00.991: Mica Modem(1/0): State Transition to Trainup
*Apr  7 12:40:02.615: Mica Modem(1/0): State Transition to EC Negotiating
*Apr  7 12:40:02.615: Mica Modem(1/3): State Transition to EC Negotiating
CONNECT 31200 /V.42/V.42bis
Router>
*Apr  7 12:40:05.983: Mica Modem(1/0): State Transition to Steady State
*Apr  7 12:40:05.983: Mica Modem(1/3): State Transition to Steady State+++
OK
ath
*Apr  7 12:40:09.167: Mica Modem(1/0): State Transition to Steady State Escape
*Apr  7 12:40:10.795: Mica Modem(1/0): State Transition to Terminating
*Apr  7 12:40:10.795: Mica Modem(1/3): State Transition to Terminating
*Apr  7 12:40:11.755: Mica Modem(1/3): State Transition to Idle
*Apr  7 12:40:11.755: Mica Modem(1/3): Went onhook
```

```
*Apr  7 12:40:11.755: CSM_PROC_IC5_OC6_CONNECTED: CSM_EVENT_MODEM_ONHOOK at slot 1, port 3
*Apr  7 12:40:11.755: VDEV_DEALLOCATE: slot 1 and port 3 is deallocated
*Apr  7 12:40:11.759: neat msg at slot 0: (0/0): Tx LOOP_OPEN (ABCD=0101)
*Apr 7 12:40:11.767: neat msg at slot 0: (1/0): Rx LOOP_OPEN (ABCD=0101)
*Apr  7 12:40:12.087: neat msg at slot 0: (1/0): Tx LOOP_OPEN (ABCD=0101)
*Apr  7 12:40:12.091: neat msg at slot 0: (0/0): Rx LOOP_OPEN (ABCD=0101)
*Apr  7 12:40:12.111: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_CALL_IDLE at slot 1 and
port 0
*Apr  7 12:40:12.111: CSM_PROC_IC5_OC6_CONNECTED: CSM_EVENT_DSX0_DISCONNECTED at slot 1,
port 0
*Apr  7 12:40:12.111: Mica Modem(1/0): Link Terminate(0x6)
*Apr  7 12:40:12.779: Mica Modem(1/3): State Transition to Terminating
*Apr  7 12:40:12.839: Mica Modem(1/3): State Transition to Idle
*Apr  7 12:40:13.495: Mica Modem(1/0): State Transition to Idle
*Apr  7 12:40:13.495: Mica Modem(1/0): Went onhook
*Apr  7 12:40:13.495: CSM_PROC_IC6_OC8_DISCONNECTING: CSM_EVENT_MODEM_ONHOOK at slot 1,
port 0
*Apr  7 12:40:13.495: VDEV_DEALLOCATE: slot 1 and port 0 is deallocated
Router#disc
Closing connection to 1.19.36.7 [confirm]
Router#
*Apr  7 12:40:18.783: Mica Modem(1/0): State Transition to Terminating
*Apr  7 12:40:18.843: Mica Modem(1/0): State Transition to Idle
Router#
```

The MICA technologies modem goes through the following internal link states when the call comes in:

- Call Setup
- Off Hook
- Connect
- Link
- Trainup
- EC Negotiation
- Steady State

The following section describes the CSM activity for an incoming call.

When a voice call comes in, CSM is informed of the incoming call. This allocates the modem and sends the Call Setup message to the MICA modem. The Call_Proc message is sent through D channel. The modem sends an offhook message to CSM by sending the state change to Call Setup. The D channel then sends a CONNECT message. When the CONNECT_ACK message is received, the Link initiate message is sent to the MICA modem and it negotiates the connection with the remote modem. In the following debug examples, a modem on slot 1, port 13 is allocated. It goes through its internal states before it is in Steady State and answers the call.

```
Router# debug modem csm

Modem Management Call Switching Module debugging is on
*May 13 15:01:00.609: MODEM_REPORT:dchan_idb=0x60D437F8, call_id=0xE, ces=0x1
   bchan=0x12, event=0x1, cause=0x0
*May 13 15:01:00.609: VDEV_ALLOCATE: slot 1 and port 13 is allocated.
*May 13 15:01:00.609: MODEM_REPORT(000E): DEV_INCALL at slot 1 and port 13
*May 13 15:01:00.609: CSM_PROC_IDLE: CSM_EVENT_ISDN_CALL at slot 1, port 13
*May 13 15:01:00.609: Mica Modem(1/13): Configure(0x0)
*May 13 15:01:00.609: Mica Modem(1/13): Configure(0x0)
*May 13 15:01:00.609: Mica Modem(1/13): Configure(0x6)
*May 13 15:01:00.609: Mica Modem(1/13): Call Setup
*May 13 15:01:00.661: Mica Modem(1/13): State Transition to Call Setup
*May 13 15:01:00.661: Mica Modem(1/13): Went offhook
```

```
*May 13 15:01:00.661: CSM_PROC_IC1_RING: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 13
*May 13 15:01:00.661: MODEM_REPORT:dchan_idb=0x60D437F8, call_id=0xE, ces=0x1
   bchan=0x12, event=0x4, cause=0x0
*May 13 15:01:00.661: MODEM_REPORT(000E): DEV_CONNECTED at slot 1 and port 13
*May 13 15:01:00.665: CSM_PROC_IC3_WAIT_FOR_CARRIER:
CSM_EVENT_ISDN_CONNECTED at slot 1, port 13
*May 13 15:01:00.665: Mica Modem(1/13): Link Initiate
*May 13 15:01:00.693: Mica Modem(1/13): State Transition to Connect
*May 13 15:01:01.109: Mica Modem(1/13): State Transition to Link
*May 13 15:01:09.433: Mica Modem(1/13): State Transition to Trainup
*May 13 15:01:11.541: Mica Modem(1/13): State Transition to EC Negotiating
*May 13 15:01:12.501: Mica Modem(1/13): State Transition to Steady State
```

The following section describes the status of CSM when a call is connected.

The **show modem csm x/y** command is similar to AS5200 access server. For an active incoming analog call, the modem_status and csm_status should be VDEV_STATUS_ACTIVE_CALL and CSM_IC4_CONNECTED, respectively.

```
Router# show modem csm 1/13

MODEM_INFO: slot 1, port 13, unit 0, modem_mask=0x0000, modem_port_offset=0
tty_hwidb=0x60D0BCE0, modem_tty=0x60B6FE7C, oobp_info=0x00000000,
modem_pool=0x60ADC998
modem_status(0x0002):VDEV_STATUS_ACTIVE_CALL.
csm_state(0x0204)=CSM_IC4_CONNECTED, csm_event_proc=0x600C6968, current
call thru PRI line
invalid_event_count=0, wdt_timeout_count=0
wdt_timestamp_started is not activated
wait_for_dialing:False, wait_for_bchan:False
pri_chnl=TDM_PRI_STREAM(s0, u0, c18), modem_chnl=TDM_MODEM_STREAM(s1, c13)
dchan_idb_start_index=0, dchan_idb_index=0, call_id=0x000E, bchan_num=18
csm_event=CSM_EVENT_ISDN_CONNECTED, cause=0x0000
ring_indicator=0, oh_state=0, oh_int_enable=0, modem_reset_reg=0
ring_no_answer=0, ic_failure=0, ic_complete=1
dial_failure=0, oc_failure=0, oc_complete=0
oc_busy=0, oc_no_dial_tone=0, oc_dial_timeout=0
remote_link_disc=0, stat_busyout=0, stat_modem_reset=0
oobp_failure=0
call_duration_started=1d02h, call_duration_ended=00:00:00,
total_call_duration=00:00:00
The calling party phone number = 4085552400
The called party phone number  = 4085551400
total_free_rbs_timeslot = 0, total_busy_rbs_timeslot = 0,
total_dynamic_busy_rbs_timeslot = 0, total_static_busy_rbs_timeslot = 0,
min_free_modem_threshold = 6
```

The following section describes the CSM activity for an outgoing call.

For MICA modems, the dial tone is not required to initiate an outbound call. Unlike in the AS5200, the digit collection step is not required. The dialed digit string is sent to the CSM in the outgoing request to the CSM. CSM signals the D channel to generate an outbound voice call, and the B channel assigned is connected to the modem and the CSM.

The modem is ordered to connect to the remote side with a CONNECT message, and by sending a link initiate message, the modem starts to train.

```
Router# debug modem csm

Modem Management Call Switching Module debugging is on
Router# debug isdn q931
ISDN Q931 packets debugging is on
*May 15 12:48:42.377: Mica Modem(1/0): Rcvd Dial String(5552400)
*May 15 12:48:42.377: CSM_PROC_IDLE: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0
*May 15 12:48:42.377: CSM_PROC_OC3_COLLECT_ALL_DIGIT:
CSM_EVENT_GET_ALL_DIGITS at slot 1, port 0
*May 15 12:48:42.377: CSM_PROC_OC3_COLLECT_ALL_DIGIT: called party num:
(5552400) at slot 1, port 0
*May 15 12:48:42.381:  process_pri_call making a voice_call.
*May 15 12:48:42.381: ISDN Se0:23: TX ->  SETUP pd = 8  callref = 0x0011
*May 15 12:48:42.381:         Bearer Capability i = 0x8090A2
*May 15 12:48:42.381:         Channel ID i = 0xE1808397
*May 15 12:48:42.381:         Called Party Number i = 0xA1, '5552400'
*May 15 12:48:42.429: ISDN Se0:23: RX <-  CALL_PROC pd = 8  callref = 0x8011
*May 15 12:48:42.429:         Channel ID i = 0xA98397
*May 15 12:48:42.429: MODEM_REPORT:dchan_idb=0x60D437F8, call_id=0xA011, ces=0x1
   bchan=0x16, event=0x3, cause=0x0
*May 15 12:48:42.429: MODEM_REPORT(A011): DEV_CALL_PROC at slot 1 and port 0
*May 15 12:48:42.429: CSM_PROC_OC4_DIALING: CSM_EVENT_ISDN_BCHAN_ASSIGNED
at slot 1, port 0
*May 15 12:48:42.429: Mica Modem(1/0): Configure(0x1)
*May 15 12:48:42.429: Mica Modem(1/0): Configure(0x0)
*May 15 12:48:42.429: Mica Modem(1/0): Configure(0x6)
*May 15 12:48:42.429: Mica Modem(1/0): Call Setup
*May 15 12:48:42.489: Mica Modem(1/0): State Transition to Call Setup
*May 15 12:48:42.589: ISDN Se0:23: RX <-  ALERTING pd = 8  callref = 0x8011
*May 15 12:48:43.337: ISDN Se0:23: RX <-  CONNECT pd = 8  callref = 0x8011
*May 15 12:48:43.341: MODEM_REPORT:dchan_idb=0x60D437F8, call_id=0xA011, ces=0x1
   bchan=0x16, event=0x4, cause=0x0
*May 15 12:48:43.341: MODEM_REPORT(A011): DEV_CONNECTED at slot 1 and port 0
*May 15 12:48:43.341: CSM_PROC_OC5_WAIT_FOR_CARRIER:
CSM_EVENT_ISDN_CONNECTED at slot 1, port 0
*May 15 12:48:43.341: Mica Modem(1/0): Link Initiate
*May 15 12:48:43.341: ISDN Se0:23: TX ->  CONNECT_ACK pd = 8  callref = 0x0011
*May 15 12:48:43.385: Mica Modem(1/0): State Transition to Connect
*May 15 12:48:43.849: Mica Modem(1/0): State Transition to Link
*May 15 12:48:52.665: Mica Modem(1/0): State Transition to Trainup
*May 15 12:48:54.661: Mica Modem(1/0): State Transition to EC Negotiating
*May 15 12:48:54.917: Mica Modem(1/0): State Transition to Steady State
```

Related Commands

Command	Description
debug modem oob	Creates modem startup messages between the network management software and the modem on the specificed OOB port.
debug modem trace	Performs a call trace on the specified modem, which allows you to determine why calls are terminated.

debug modem dsip

To display output for modem control messages that are received or sent to the router, use the **debug modem dsip** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem dsip {*tty-range* | **group** | *shelf*/*slot*/*port*}

no debug modem dsip {*tty-range* | **group** | *shelf*/*slot*/*port*}

Syntax Description

tty-range	Modem tty number or range. You can specify a single TTY line number or a range from 0 through the number of modems you have in your Cisco AS5800 access server. Be sure to include a dash (-) between the range values you specify.
group	Modem group information.
shelf/slot/port	Location of the modem by shelf/slot/port numbers for internal modems.

Command Modes

Privileged EXEC

CommandHistory

Release	Modification
11.3(2)AA	This command was introduced.

Usage Guidelines

The **debug modem dsip** command displays each DSIP message that relates to a modem and is sent from or received at the router shelf. This command can be applied to a single modem or a group of modems.

Examples

The following examples show a display of the available **debug modem** command options and **debug modem dsip** command options:

```
Router# debug modem ?

  dsip         Modem DSIP activity
  maintenance  Modem maintenance activity
  oob          Modem out of band activity
  trace        Call Trace Upload
  traffic      Modem data traffic
  <cr>

Router# debug modem dsip ?

  <0-935>  First Modem TTY Number
  group    Modem group information
  x/y/z    Shelf/Slot/Port for Internal Modems
  <cr>
```

The following example indicates that an RTS status message was received from the router shelf, and an ACK message was sent back:

```
Router# debug modem dsip

00:11:02: RSMODEM_SEND-1/2/06: MODEM_RING_INDICATION_MSG cci1 si0 ms0 mm65535,0 dc0
00:11:02: RSMODEM_sRCV-1/2/06:l12,MODEM_CALL_ACK_MSG:
00:11:02: RSMODEM_SEND-1/2/06: MODEM_CALL_ACCEPT_MSG
00:11:11: RSMODEM_sRCV-2:l0,MODEM_POLL_MSG: 0 16 0 7 0 146 0 36 21
00:11:18: RSMODEM_sRCV-1/2/06:l12,MODEM_SET_DCD_STATE_MSG: 1
00:11:19: RSMODEM_SEND-1/2/06: MODEM_RTS_STATUS_MSG 1
00:11:19: RSMODEM_dRCV-2:l1258607996,MODEM_RTS_STATUS_MSG: 0 6 0 23 0 0 0 0 0
00:11:23: RSMODEM_sRCV-2:l0,MODEM_POLL_MSG: 0 16 0 7 0 146 0 150 21
00:12:31: RSMODEM_sRCV-1/2/06:l12,MODEM_SET_DCD_STATE_MSG: 0
00:12:31: RSMODEM_SEND-1/2/06: MODEM_CALL_HANGUP_MSG
00:12:31: RSMODEM_sRCV-1/2/06:l12,MODEM_ONHOOK_MSG:
00:12:32: RSMODEM_SEND-1/2/06: MODEM_RTS_STATUS_MSG 1
00:12:32: RSMODEM_SEND-1/2/06: MODEM_SET_DTR_STATE_MSG 0
00:12:32: RSMODEM_dRCV-2:l1258659676,MODEM_RTS_STATUS_MSG: 0 6 0 16 0 0 0 0 0
00:12:32: RSMODEM_SEND-1/2/06: MODEM_RTS_STATUS_MSG 1
00:12:32: RSMODEM_dRCV-2:l1258600700,MODEM_RTS_STATUS_MSG: 0 6 0 13 0 0 0 0 0
00:12:33: RSMODEM_SEND-1/2/06: MODEM_SET_DTR_STATE_MSG 0
00:12:33: RSMODEM_SEND-1/2/06: MODEM_RTS_STATUS_MSG 1
00:12:33: RSMODEM_dRCV-2:l1258662108,MODEM_RTS_STATUS_MSG: 0 6 0 16 0 0 0 0 0
00:12:35: RSMODEM_sRCV-2:l0,MODEM_POLL_MSG: 0 16 0 7 0 146 1 34 22
00:12:38: RSMODEM_SEND-1/2/06: MODEM_SET_DTR_STATE_MSG 1
00:12:47: RSMODEM_sRCV-2:l0,MODEM_POLL_MSG: 0 16 0 7 0 146 0 12 22
```

Table 159 describes the significant fields shown in the display.

Table 159 debug modem dsip Field Descriptions

Field	Description
RSMODEM_SEND-1/2/06	Router shelf modem shelf sends a MODEM_RING_INDICATION_MSG message.
RSMODEM_sRCV-1/2/06	Router shelf modem received a MODEM_CALL_ACK_MSG message.
MODEM_CALL_ACCEPT_MSG	Router shelf accepts the call.
MODEM_CALL_HANGUP_MSG	Router shelf sends a hangup message.
MODEM_RTS_STATUS_MSG	Request to send message status.

Related Commands

Command	Description
debug modem traffic	Displays output for framed, unframed, and asynchronous data transmission received from the modem cards.
debug dsip	Displays output for DSIP used between the router shelf and the dial shelf.

debug modem oob

To debug the out-of-band port used to poll modem events on the modem, use the **debug modem oob** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem oob [*slot/modem-port* | **group** *group-number*]

no debug modem oob [*slot/modem-port* | **group** *group-number*]

Syntax Description

slot/modem-port	(Optional) The slot and modem port number.
group *group-number*	(Optional) The modem group.

Command Modes

Privileged EXEC

Usage Guidelines

The message types and sequence numbers that appear in the debug output are initiated by the Modem Out-of-Band Protocol and used by service personnel for debugging purposes.

Caution Entering the **debug modem oob** command without specifying a slot and modem number debugs *all* out-of-band ports, which generates a substantial amount of information.

Examples

The following is sample output from the **debug modem oob** command. This example debugs the out-of-band port on modem 2/0, which creates modem startup messages between the network management software and the modem.

```
Router# debug modem oob 2/0

MODEM(2/0): One message sent --Message type:3, Sequence number:0
MODEM(2/0): Modem DC session data reply
MODEM(2/0): One message sent --Message type:83, Sequence number:1
MODEM(2/0): DC session event =
MODEM(2/0): One message sent --Message type:82, Sequence number:2
MODEM(2/0): No status changes since last polled
MODEM(2/0): One message sent --Message type:3, Sequence number:3
MODEM(2/0): Modem DC session data reply
MODEM(2/0): One message sent --Message type:83, Sequence number:4
```

debug modem relay errors

To view modem relay network errors, use the **debug modem relay errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **errors**

no debug modem relay [**call-identifier** *call-setup-time call-index*] **errors**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers and the Cisco AS5300 universal access server.

Usage Guidelines

In a stable modem relay network, the **debug modem relay errors** command produces little output.

Examples

A sample output of the **debug modem relay errors** command is shown below.

The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 28 tm 11944 OUT ERR, pb=12, payload: 00 06
00 00 00 00 00 07 00 00 01 DE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 29 tm 11944 OUT ERR, pb=12, payload: 00
06 00 00 00 00 00 04 00 00 00 BE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 30 tm 11944 OUT ERR, pb=12, payload: 00
06 00 00 00 00 00 05 FF FF FF FD
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay events	Displays events that may cause failure of the modem relay network.

debug modem relay events

To view the events that may cause failure of the modem relay network, use the **debug modem relay events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **events**

no debug modem relay [**call-identifier** *call-setup-time call-index*] **events**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers and the Cisco AS5300 universal access server.

Usage Guidelines

In a stable modem relay network, the **debug modem relay events** command produces little output.

Examples

A sample output of the **debug modem relay events** command is shown below.

The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 28 tm 11944 OUT EVNT, pb=12, payload: 00
06 00 00 00 00 00 07 00 00 01 DE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 29 tm 11944 OUT EVNT, pb=12, payload: 00
06 00 00 00 00 00 04 00 00 00 BE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 30 tm 11944 OUT EVNT, pb=12, payload: 00
06 00 00 00 00 00 05 FF FF FF FD
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay packetizer

To view events occurring in the modem relay packetizer module, use the **debug modem relay packetizer** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **packetizer**

no debug modem relay [**call-identifier** *call-setup-time call-index*] **packetizer**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay packetizer** command. Using the **debug modem relay packetizer** command generates a large volume of debugs, which can affect router performance.

Examples

A sample output of the **debug modem relay packetizer** command is shown below.

The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:33:33.715:ModemRelay pkt[0:D:11]. sqn 8 tm 47610 IN PKTZR, pb=7, payload: 82 38
00 18 03 01 87
*Jan 11 05:33:33.727:ModemRelay pkt[0:D:11]. sqn 9 tm 47616 OUT PKTZR, pb=7, payload: 82
20 00 18 03 01 47
*Jan 11 05:33:35.719:ModemRelay pkt[0:D:11]. sqn 10 tm 49614 IN PKTZR, pb=7, payload: 82
39 00 18 03 01 87
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay physical

To view modem relay physical layer packets, use the **debug modem relay physical** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **physical**

no debug modem relay [**call-identifier** *call-setup-time call-index*] **physical**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay physical** command. Using the **debug modem relay physical** command generates a large volume of debugs, which can affect router performance.

Examples

A sample output of the **debug modem relay physical** command is shown below.

The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 28 tm 11944 OUT PHYS, pb=12, payload: 00
06 00 00 00 00 00 07 00 00 01 DE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 29 tm 11944 OUT PHYS, pb=12, payload: 00
06 00 00 00 00 00 04 00 00 00 BE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 30 tm 11944 OUT PHYS, pb=12, payload: 00
06 00 00 00 00 00 05 FF FF FF FD
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay sprt

To view modem relay Simple Packet Relay Transport (SPRT) protocol packets, use the **debug modem relay sprt** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **sprt**

no debug modem relay [**call-identifier** *call-setup-time call-index*] **sprt**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay sprt** command. Using the **debug modem relay sprt** command generates a large volume of debugs, which can affect router performance.

Examples

A sample output of the **debug modem relay sprt** command is shown below.

The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:37:16.151:ModemRelay pkt[0:D:11]. sqn 34 tm 7910 OUT SPRT, pb=4, payload: 02 00
03 71
*Jan 11 05:37:16.295:ModemRelay pkt[0:D:11]. sqn 35 tm 8048 IN SPRT, pb=13, payload: 02 00
01 F1 F7 7E FD F5 90 F3 3E 90 55
*Jan 11 05:37:16.303:ModemRelay pkt[0:D:11]. sqn 36 tm 8060 IN SPRT, pb=6, payload: 02 00
01 41 04 00
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay udp

To view events occurring in the User Datagram Protocol (UDP) stack, use the **debug modem relay udp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **udp**

no debug modem relay [**call-identifier** *call-setup-time call-index*] **udp**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay udp** command. Using the **debug modem relay udp** command generates a large volume of debugs, which can affect router performance.

Examples

A sample output of the **debug modem relay udp** command is shown below.

The output shows three UDP packets related to modem relay. In the sample output, OUT or IN represent packet direction, and UDP indicates the specific layer that reported the packet.

```
Jan 1 03:39:29.407:ModemRelay pkt[0:D (4)]. sqn 61 tm 3060 OUT UDP, pb=6, payload: 80 00
00 00 00 00
*Jan 1 03:39:29.471:ModemRelay pkt[0:D (4)]. sqn 62 tm 3120 IN UDP, pb=6, payload: 40 00
00 00 00 00
*Jan 1 03:39:29.471:ModemRelay pkt[0:D (4)]. sqn 63 tm 3120 IN UDP, pb=6, payload: 80 00
00 00 00 00
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay v42

To view events occurring in the V.42 layer, use the **debug modem relay v42** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **v42**

no debug modem relay [**call-identifier** *call-setup-time call-index*] **v42**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay v42** command. Using the **debug modem relay v42** command generates a large volume of debugs, which can affect router performance.

Examples

A sample output of the **debug modem relay v42** command is shown below.

The output shows the sequence number of the packet, timestamp, direction, layer, and payload-bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:42:08.715:ModemRelay pkt[0:D:13]. sqn 3 tm 10104 OUT V42, pb=43, payload: 03 AF
82 80 00 13 03 03 8A 89 00 05 02 03 E0 06 02 03 E0 07 01 08 08 01 08 F0 00 0F 00 03 56 34
32 01 01 03 02 02 04 00 03 01 20
*Jan 11 05:42:08.847:ModemRelay pkt[0:D:13]. sqn 4 tm 10236 IN V42, pb=2, payload: 03 7F
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem trace

To debug a call trace on the modem to determine why calls are terminated, use the **debug modem trace** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem trace [**normal** | **abnormal** | **all**] [*slot/modem-port* | **group** *group-number*]

no debug modem trace [**normal** | **abnormal** | **all**] [*slot/modem-port* | **group** *group-number*]

Syntax Description

normal	(Optional) Uploads the call trace to the syslog server on normal call termination (for example, a local user hangup or a remote user hangup).
abnormal	(Optional) Uploads the call trace to the syslog server on abnormal call termination (for example, any call termination other than normal termination, such as a lost carrier or a watchdog timeout).
all	(Optional) Uploads the call trace on all call terminations including normal and abnormal call termination.
slot/modem-port	(Optional) The slot and modem port number.
group *group-number*	(Optional) The modem group.

Command Modes

Privileged EXEC

Usage Guidelines

The **debug modem trace** command applies only to manageable modems. For additional information, use the **show modem** command.

Examples

The following is sample output from the **debug modem trace abnormal** command:

```
Router# debug modem trace abnormal 1/14

Modem 1/14 Abnormal End of Connection Trace. Caller 123-4567
    Start-up Response: AS5200 Modem, Firmware 1.0
    Control Reply: 0x7C01
    DC session response: brasil firmware 1.0
    RS232 event:
    DSR=On, DCD=On, RI=Off, TST=Off
    changes: RTS=No change, DTR=No change, CTS=No change
    changes: DSR=No change, DCD=No change, RI=No change, TST=No change
    Modem State event: Connected
    Connection event: Speed = 19200, Modulation = VFC
    Direction = Originate, Protocol = reliable/LAPM, Compression = V42bis
    DTR event: DTR On
    Modem Activity event: Data Active
    Modem Analog signal event: TX = -10, RX = -24, Signal to noise = -32
    End connection event: Duration = 10:34-11:43,
    Number of xmit char =    67, Number of rcvd char = 88, Reason: Watchdog Time-out.
```

Related Commands

Command	Description
debug modem csm	Debugs the CSM used to connect calls on the modem.
debug modem oob	Creates modem startup messages between the network management software and the modem on the specificed OOB port.

debug modem traffic

To display output for framed, unframed, and asynchronous data sent received from the modem cards, use the **debug modem traffic** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem traffic

no debug modem traffic

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(2)AA	This command was introduced.

Usage Guidelines

The **debug modem traffic** command displays output for framed, unframed, and asynchronous data sent or received by the modem cards.

Examples

The following example displays information about unframed or framed data sent to or received from the modem cards:

```
Router# debug modem traffic

MODEM-RAW-TX:modem = 6/5/00, length = 1, data = 0x61, 0xFF, 0x7D, 0x23
MODEM-RAW-RX:modem = 6/5/00, length = 1, data = 0x61, 0x0, 0x0, 0x0
```

The information indicates unframed asynchronous data transmission and reception involving the modem on shelf 6, slot 5, port 00.

The following example displays framed asynchronous data transmission and reception involving the modem on shelf 6, slot 5, port 00:

```
Router# debug modem traffic

MODEM-FRAMED-TX:modem = 6/5/00, length = 8, data = 0xFF, 0x3, 0x82
MODEM-FRAMED-RX:modem = 6/5/00, length = 14, data = 0xFF, 0x3, 0x80
```

Related Commands

Command	Description
debug modem dsip	Displays output for modem control messages that are received or sent to the router.

debug mpls adjacency

To display changes to label switching entries in the adjacency database, use the **debug mpls adjacency** EXEC command. To disable debugging output, use the **no** form of this command.

debug mpls adjacency

no debug mpls adjacency

Usage Guidelines

This command has no keywords or arguments.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.1(3)T	This command was modified to reflect new MPLS IETF terminology and CLI command syntax.

Usage Guidelines

Use the **debug mpls adjacency** command to monitor when entries are updated in or added to the adjacency database.

Examples

The following is sample output generated by the **debug mpls adjacency** command:

```
Router# debug mpls adjacency

TAG ADJ: add 10.10.0.1, Ethernet0/0/0
TAG ADJ: update 10.10.0.1, Ethernet0/0/0
```

Table 160 describes the significant fields shown in the sample display above.

Table 160 debug mpls adjacency Command Field Description

Field	Description
add	Adding an entry to the database.
update	Updating the MAC address for an existing entry.
10.10.0.1	Address of neighbor TSR.
Ethernet0/0/0	Connecting interface.

debug mpls atm-cos

To display ATM label VC bind or request activity that is based on the configuration of a Quality of Service (QoS) map, use the **debug mpls atm-cos** command in privileged EXEC mode.

debug mpls atm-cos [*bind* | *request*]

Syntax Description

bind	Specifies debug information about bind responses for a VC path.
request	Specifies debug information about bind requests for a VC path.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF syntax and terminology.
12.2(2)T	This command was incorporated into Cisco IOS Release 12.2(2)T.

Examples

The following command sequence demonstrates how to obtain sample output from the **debug mpls atm-cos** command.

First, display the MPLS forwarding table to see which prefixes are associated with a single LVC, as shown below:

```
Router# show mpls forwarding

Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
26     28          17.17.17.17/32   0          PO6/0      point2point
27     Pop tag     11.11.11.11/32   1560       PO6/0      point2point
28     27          16.16.16.16/32   0          PO6/0      point2point
29     30          92.0.0.0/8       0          PO6/0      point2point
30     Pop tag     95.0.0.0/8       2600       PO6/0      point2point
31     2/34        10.10.10.10/32   0          AT2/0.1    point2point
32     Pop tag     14.14.14.14/32   0          Fa5/0      91.0.0.1
33     Pop tag     90.0.0.0/8       0          Fa5/0      91.0.0.1
34     Pop tag     96.0.0.0/8       0          Fa5/0      91.0.0.1
       2/36        96.0.0.0/8       0          AT2/0.1    point2point
35     35          93.0.0.0/8       0          PO6/0      point2point
36     36          12.12.12.12/32   0          PO6/0      point2point
37     37          15.15.15.15/32   0          PO6/0      point2point
38     37          18.18.18.18/32   0          Fa5/0      91.0.0.1
39     39          97.0.0.0/8       540        PO6/0      point2point
40     40          98.0.0.0/8       0          PO6/0      point2point
```

Second, enable debugging of request and bind events, as shown in the command sequence below:

```
Router# debug mpls atm-cos ?
  bind     Bind response for VC path
  request  Requests for VC binds path

Router# debug mpls atm-cos request
ATM TAGCOS VC requests debugging is on

Router# debug mpls atm-cos bind
ATM TAGCOS Bind response debugging is on
```

Third, configure an MPLS ATM subinterface for multi-VC mode. The corresponding request and bind events are displayed, as shown below:

```
Router# conf terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# int a2/0.1
Router(config-subif)# mpls atm multi-vc
Router(config-subif)# end
Router#
19:59:14:%SYS-5-CONFIG_I:Configured from console by console
Router#
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, available
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, standard
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, premium
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, control
19:59:24:TAGCOS-REQ:vc request 96.0.0.0/8, available
19:59:24:TAGCOS-REQ:vc request 96.0.0.0/8, standard
19:59:24:TAGCOS-REQ:vc request 96.0.0.0/8, premium
19:59:24:TAGCOS-REQ:vc request 96.0.0.0/8, control
TAGCOS-REQ/TCATM:11.11.11.11/32,len=4352,band=1099528405504,class=0x700
TAGCOS-REQ/TCATM:12.12.12.12/32,len=4352,band=2199040033280,class=0x700
TAGCOS-REQ/TCATM:13.13.13.13/32,len=4352,band=3298551661056,class=0x700
TAGCOS-REQ/TCATM:14.14.14.14/32,len=4352,band=4398063288832,class=0x700
TAGCOS-REQ/TCATM:15.15.15.15/32,len=4352,band=5497574916608,class=0x700
TAGCOS-REQ/TCATM:16.16.16.16/32,len=4352,band=6597086544384,class=0x700
TAGCOS-REQ/TCATM:17.17.17.17/32,len=4352,band=7696598172160,class=0x700
TAGCOS-REQ/TCATM:18.18.18.18/32,len=4352,band=8796109799936,class=0x700
TAGCOS-REQ/TCATM:90.0.0.0/8,len=768,band=3940649674539009,class=0x2
TAGCOS-REQ/TCATM:91.0.0.0/8,len=768,band=3940649674604545,class=0x2
TAGCOS-REQ/TCATM:92.0.0.0/8,len=768,band=3940649674670081,class=0x2
TAGCOS-REQ/TCATM:93.0.0.0/8,len=768,band=3940649674735617,class=0x2
TAGCOS-REQ/TCATM:94.0.0.0/8,len=768,band=3940649674801153,class=0x2
TAGCOS-REQ/TCATM:95.0.0.0/8,len=768,band=3940649674866689,class=0x2
TAGCOS-REQ/TCATM:97.0.0.0/8,len=768,band=3940649674932225,class=0x2
TAGCOS-REQ/TCATM:98.0.0.0/8,len=768,band=3940649674997761,class=0x2
TAGCOS-BIND:binding_ok 10.10.10.10/32,VCD=41 - control 41,41,41,41
TAGCOS-BIND:binding_ok 10.10.10.10/32, Inform TFIB pidx=0, in_tag=31, idx=0x80000000
TAGCOS-BIND:binding_ok 96.0.0.0/8,VCD=42 - control 42,42,42,42
TAGCOS-BIND:binding_ok 96.0.0.0/8, Inform TFIB pidx=1, in_tag=34, idx=0x80000001
TAGCOS-BIND:binding_ok 10.10.10.10/32,VCD=43 - premium 43,43,43,41
TAGCOS-BIND:binding_ok 96.0.0.0/8,VCD=44 - premium 44,44,44,42
TAGCOS-BIND:binding_ok 10.10.10.10/32,VCD=45 - standard 45,45,43,41
TAGCOS-BIND:binding_ok 96.0.0.0/8,VCD=46 - standard 46,46,44,42
TAGCOS-BIND:binding_ok 10.10.10.10/32,VCD=47 - available 47,45,43,41
TAGCOS-BIND:binding_ok 96.0.0.0/8,VCD=48 - available 48,46,44,42
```

debug mpls atm-ldp api

To display information about the VCI allocation of label VCs (LVCs), label-free requests, and cross-connect requests, use the **debug mpls atm-ldp api** command in privileged EXEC mode. To disable debugging output, use the **no** form of the command.

debug mpls atm-ldp api

no debug mpls atm-ldp api

Syntax Description

This command has no keywords or arguments.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.1(2)T	This command was modified.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

Use the **debug mpls atm-ldp api** command in conjunction with the **debug mpls atm-ldp routes** and **debug mpls atm-ldp states** commands to display more complete information about an LVC.

Examples

The following shows sample output from the **debug mpls atm-ldp api** command:

```
Router# debug mpls atm-ldp api

Tailend Router Free label Req 167.50.0.0 on ATM0/0.2 VPI/VCI 1/674
 TAGATM_API: received label free request
             interface: ATM0/0.2  dir: in  vpi: 1  vci: 674
 TAGATM_API: completed label free
             interface: ATM0/0.2  vpi: 1  vci: 674
             result: TAGATM_OK
```

Table 161 describes the significant fields shown in the display above.

Table 161 debug mpls atm-ldp api Field Descriptions

Field	Description
TAGATM_API	The subsystem that displays the message.
interface	The interface used by the driver to allocate or free VPI/VCI resources.
dir	The direction of the VC: • In—Input or receive VC • Out—Output VC
vpi	Virtual path identifier.
vci	Virtual channel identifier.
result	The return error code from the driver API.

Related Commands

Command	Description
debug mpls atm-ldp states	Displays information about LVC state transitions as they occur.

debug mpls atm-ldp failure

To display failure information about the label-controlled asynchronous transfer mode (LC-ATM), use the **debug mpls atm-ldp failure** command in privileged EXEC mode. To disable debugging output, use the **no** form of the command.

debug mpls atm-ldp failure

no debug mpls atm-ldp failure

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(8)T	This command was introduced.

Usage Guidelines

Use the **debug mpls atm-ldp failure** command to display failure information about the LC-ATM. This command is useful for determining failure cases. This command displays only failure information, unlike the **debug mpls atm-ldp api** command, which displays all application programming interface (API) events.

Examples

The following shows sample output from the **debug mpls atm-ldp failure** command:

The following failure message displays during a race condition where the LC-ATM attempts to allocate LVCs on an interface where MPLS has been disabled:

```
Router# debug mpls atm-ldp failure

TAGATM_API_FAILURE: allocate_tag_req on ATM1/0/0 tagsw not enabled
```

The following failure message displays when the LC-ATM fails to deallocate the output leg Label Virtual Circuit (LVC) of a cross connect:

```
Router# debug mpls atm-ldp failure

TAGATM_API_FAILURE: connDeAllocateHalfLeg returned false interface: ATM1/0/0
vpi: 1  vci: 48
```

The following failure message displays when a cross connect cannot be installed on the switching fabric. The result code is also provided.

```
Router# debug mpls atm-ldp failure

TAGATM_API_FAILURE: setup_xconn_req InstallSvcXconn failed result
```

The following message displays when attempts to establish a cross connect fail. The result describes the reason for the failure.

```
Router# debug mpls atm-ldp failure

TCATM-4-XCONNECT_FAILED: 10.254.13.237/32 for ATM0/1/2 ATM1/0/0
TAGATM_API: x-conn setup request completed
             input interface: ATM0/1/2  vpi: 1  vci: 48
            output interface: ATM1/0/0  vpi: 2  vci: 2038
            result = TAGATM_FAIL
Xconnect setup response for 10.254.13.215: failure, 8
```

The following message displays when attempts to remove a cross connect fail. The result describes why the cross connect cannot be removed.

```
Router# debug mpls atm-ldp failure

TCATM-4-XCONNECT_REMOVE_FAILED: Remove XConnect API failed for ATM1/0/12 1/894
-> ATM1/0/13 1/528
TAGATM_API: x-conn remove request completed
             input interface: ATM1/0/12  vpi: 1  vci: 894
            output interface: ATM1/0/13  vpi: 1  vci: 528
            result = TAGATM_FAIL
```

Related Commands

Command	Description
debug mpls atm-ldp api	Displays all driver API events.

debug mpls atm-ldp routes

Tto display information about the state of the routes for which VCI requests are being made, use the **debug mpls atm-ldp routes** command in privileged EXEC mode. To disable debugging output, use the **no** form of the command.

debug mpls atm-ldp routes

no debug mpls atm-ldp routes

Syntax Description

This command has no keywords or arguments.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Command	Modification
11.1 CT	This command was introduced.
12.1(2)T	This command was modified.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

When there are many routes and system activities (that is, shutting down of interfaces, learning of new routes, and so forth), the **debug mpls atm-ldp routes** command displays extensive information that might interfere with system timing. Most commonly, this interference affects normal LDP operation. To avoid this problem, you can increase the LDP hold time by means of the **mpls ldp holdtime** command.

Examples

The following shows sample output from the **debug mpls atm-ldp routes** command:

```
Router# debug mpls atm-ldp routes

CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.7.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.7.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.7.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.8.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.8.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.8.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.9.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.9.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.9.0.0/16
```

```
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.10.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.10.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.10.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.11.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.11.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.11.0.0/16
```

Table 162 describes the significant fields in the display above.

Table 162 debug mpls atm-ldp routes Field Descriptions

Field	Description
CleanupRoutes	Cleans up the routing table after a route has been deleted.
not deleting route of idb ATM0/0.2	The route cleanup event has not removed the specified route.
rdbIndex	Index identifying the route.
tcatmFindRouteTags	Request a VC for the route.
idb	The internal descriptor for an interface.
nh	Next hop for the route.
index	Identifier for the route.
AddNewRoute	Action of adding routes for a prefix or address.

Related Commands

Command	Description
mpls ldp holdtime	Changes the time an LDP session will be maintained in the absence of LDP messages from the session peer.

debug mpls atm-ldp states

To display information about LVC state transitions as they occur, use the **debug mpls atm-ldp states** command in privileged EXEC mode. To disable debugging output, use the **no** form of the command.

debug mpls atm-ldp states

no debug mpls atm-ldp states

Syntax Description

This command has no keywords or arguments.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.1(2)T	This command was modified.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

When there are many routes and system activities (such as shutting down of interfaces, learning of new routes, and so forth), the **debug mpls atm-ldp states** command outputs extensive information that might interfere with system timing. Most commonly, this interference affects normal LDP operation. To avoid this problem, you should increase the LDP hold time by means of the **mpls ldp holdtime** command.

Examples

The following shows sample output from the **debug mpls atm-ldp states** command:

```
Router# debug mpls atm-ldp states

Transit Output 166.35.0.0 VPI/VCI 1/67 Active -> XmitRelease NoPath
Transit Input 166.35.0.0 VPI/VCI 1/466 Active -> ApiWaitParentLoss ParentLoss
Transit Input 166.35.0.0 VPI/VCI 1/466 ApiWaitParentLoss -> ParentWait ApiSuccess
Transit Input 166.35.0.0 VPI/VCI 1/466 ParentWait -> XmitWithdraw NoPath
Transit Input 166.35.0.0 VPI/VCI 1/466 XmitWithdraw -> XmitWithdraw Transmit
Transit Input 166.35.0.0 VPI/VCI 1/466 XmitWithdraw -> NonExistent Release
Transit Input 166.35.0.0 VPI/VCI 1/466 NonExistent -> NonExistent ApiSuccess
```

Table 163 describes the significant fields shown in the sample display above.

Table 163 debug mpls atm-ldp states Field Descriptions

Field	Description
Transit Output	Output side of a label virtual circuit (LVC).
VPI/VCI	VC value.
Transit Input	Input side of an LVC.

Related Commands

Command	Description
mpls ldp holdtime	Changes the time an LDP session is maintained in the absence of LDP messages from the session peer.

debug mpls events

To display information about significant MPLS events, use the **debug mpls events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls events

no debug mpls events

Syntax Description

This command has no keywords or arguments.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.

Usage Guidelines

Use this command to monitor significant MPLS events. For this Cisco IOS release, the only events reported by this command are changes to the MPLS router ID.

Examples

The following is sample output from the **debug mpls events** command:

```
Router# debug mpls events

MPLS events debugging is on

TAGSW: Unbound IP address, 155.0.0.55, from Router ID
TAGSW: Bound IP address, 199.44.44.55, to Router ID
```

debug mpls l2transport ipc

To display the interprocessor communication (IPC) messages exchanged between distributed platforms, such as the Cisco 12000 series router and the Cisco 7500 series routers, use the **debug mpls l2transport ipc** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport ipc

no debug mpls l2transport ipc

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

Usage Guidelines

You can issue this command either from the line card or the route processor to log Any Transport over MPLS (AToM) updates to or from line cards. This command applies only to platforms that support distributed mode.

Examples

The following is sample output from the **debug mpls l2transport ipc** command:

```
Router# debug mpls l2transport ipc

AToM ipc debugging is on
*May 27 23:56:04.699 UTC: AToM SMGR: Repopulating line card 255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1101]: Sending Imposition update to slot
255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1101]: Imposition being done on ingress
interface
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1101]: Sending disposition update to
slot 255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1101]: Distributing disposition info to
all linecards
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 701]: Sending Imposition update to slot
255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 701]: Imposition being done on ingress
interface
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 701]: Sending disposition update to slot
255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 701]: Distributing disposition info to
all linecards
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1201]: Sending Imposition update to slot
255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1201]: Imposition being done on ingress
interface
```

```
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1201]: Sending disposition update to
slot 255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1201]: Distributing disposition info to
all linecards
```

debug mpls l2transport packet

To display information about the status of Any Transport over MPLS (ATOM) switched packets, use the **debug mpls l2transport packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport packet {**data** | **error**}

no debug mpls l2transport packet {**data** | **error**}

Syntax Description

data	Displays (in hex) the AToM switched packets for imposition and disposition. This can help validate that packets are flowing between the customer edge (CE) routers. Also, you can display the packets to check the format of the data or the data itself.
error	Displays AToM switching errors, such as the reason that packets cannot be switched. This can help identify why data is not being transported.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

Usage Guidelines

Use this command sparingly, because the command output can be overwhelming.

For platforms that support distributed switching, the command displays output only for packets switched by the central route processor module. Packets switched autonomously by the linecards are not displayed. For example, packets switched by Versatile Interface Processors (VIPs) on the Cisco 7500 router are not displayed.

Examples

The following is sample output from the **debug mpls l2transport packet** commands for a PPP over MPLS configuration:

```
Router# debug mpls l2transport packet data

AToM packet data debugging is on

Router# debug mpls l2transport packet error

AToM packet errors debugging is on

Router# show debug

AToM:
  AToM packet data debugging is on
  AToM packet errors debugging is on
```

```
*Mar 24 23:29:30.495: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:30.495: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:30.495: 0F 00 88 47 00 01 10 FF 00 01 51 02 00 00 00 00
*Mar 24 23:29:30.495: 00 FD C0 01 01 01 C0 4B 41 73 F4 00 01 00 02 CC
*Mar 24 23:29:30.495: 66 51 88 B4 CE 73 39 00 00 40 00 88 03 02 00 70
*Mar 24 23:29:30.495: 23 30 00 04 3C 61 83 C0 00 06 00 06 94 CC A7 23
*Mar 24 23:29:30.495: 49 84 D8 33 17 8C F2 60 00 11 9E 80 00 50 08 08
*Mar 24 23:29:30.495: 86 69 39 98 CD E2 02 49 B8 E9 9D 0D C6 53 A1 DC
*Mar 24 23:29:30.495: DE 72 35 88 09 E7 0C 60 61 3A 1A 4D C6 71 01 4C
*Mar 24 23:29:30.495: F2 73 CC 06 DC 38 6F 33 66 83 09 C8 CA 20 05 12
*Mar 24 23:29:30.495: 49 E5 31 00 A0 E8 6D 14 88 06 E3 21 80 C3 31 E4
*Mar 24 23:29:30.495: 28 21 E4 21 69 28 A6 2D 26 8A 45 82 02 B6 FC 39
*Mar 24 23:29:30.499: D8 60 A3 62 B1 60 A5 80
*Mar 24 23:29:31.835: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:31.835: FF 03 00 FD C0 04 8A 57 FF FF FF FF FF FF FF FF
*Mar 24 23:29:31.835: FF FF FB 14 B0 00

*Mar 24 23:29:49.423: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.423: FF 03 C0 21 01 11 00 0F 03 05 C2 23 05 05 06 5F
*Mar 24 23:29:49.423: 23 35 D4
*Mar 24 23:29:49.435: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.435: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.435: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 15 00 00
*Mar 24 23:29:49.435: C0 21 01 2F 00 0F 03 05 C2 23 05 05 06 5F CC 5F
*Mar 24 23:29:49.435: E5
*Mar 24 23:29:49.435: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.435: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.435: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 15 00 00
*Mar 24 23:29:49.435: C0 21 02 11 00 0F 03 05 C2 23 05 05 06 5F 23 35
*Mar 24 23:29:49.435: D4
*Mar 24 23:29:49.443: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.443: FF 03 C0 21 02 2F 00 0F 03 05 C2 23 05 05 06 5F
*Mar 24 23:29:49.443: CC 5F E5
*Mar 24 23:29:49.447: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.447: FF 03 C2 23 01 D0 00 1C 10 45 59 13 1A 92 FD 93
*Mar 24 23:29:49.447: 01 A2 CF B6 FB 3A 04 46 93 63 65 32 2D 67 73 72
*Mar 24 23:29:49.451: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.451: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.451: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 22 00 00
*Mar 24 23:29:49.451: C2 23 01 F5 00 1C 10 F1 98 35 3F 79 F2 1A 15 10
*Mar 24 23:29:49.451: B4 C0 73 D7 B1 9F 2A 63 65 31 2D 67 73 72
*Mar 24 23:29:49.455: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.455: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.455: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 22 00 00
*Mar 24 23:29:49.455: C2 23 02 D0 00 1C 10 56 4A 32 5B 99 55 D5 CF 44
*Mar 24 23:29:49.455: FC D3 D9 3F CC 8C A8 63 65 31 2D 67 73 72
*Mar 24 23:29:49.463: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.463: FF 03 C2 23 02 F5 00 1C 10 45 84 E4 E5 DD C0 5F
*Mar 24 23:29:49.463: FD 2F 37 63 9A 3D 03 7B B9 63 65 32 2D 67 73 72
*Mar 24 23:29:49.463: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.463: FF 03 C2 23 03 D0 00 04
*Mar 24 23:29:49.471: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.471: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.471: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 0A 00 00
*Mar 24 23:29:49.471: C2 23 03 F5 00 04
*Mar 24 23:29:49.471: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.471: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
```

```
*Mar 24 23:29:49.471: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 10 00 00
*Mar 24 23:29:49.471: 80 21 01 0B 00 0A 03 06 78 01 01 78
*Mar 24 23:29:49.475: ATOM-PPP Switching: check features failed.
```

debug mpls l2transport signaling

To display information about the Any Transport over MPLS (AToM) signaling protocol, use the **debug mpls l2transport signaling** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport signaling {**event** | **message**}

no debug mpls l2transport signaling {**event** | **message**}

Syntax Description

event	Displays AToM signaling events.
message	Displays AToM signaling status messages.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

Examples

The following is sample output from the **debug mpls l2transport signaling** command:

```
Router# debug mpls l2transport signaling event

AToM LDP event debugging is on

Router# debug mpls l2transport signaling message

AToM LDP message debugging is on

Router# show debugging

AToM:
  AToM LDP event debugging is on
  AToM LDP message debugging is on

*Mar 24 23:10:55.611: AToM LDP [9.9.9.9]: Allocate LDP instance
*Mar 24 23:10:55.611: AToM LDP [9.9.9.9]: Opening session, 1 clients
*Mar 24 23:10:56.063: %SYS-5-CONFIG_I: Configured from console by console
*Mar 24 23:10:56.583: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed
state to up
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Session is up
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Peer address change, add 1.1.1.100
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Peer address change, add 46.1.1.6
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Peer address change, add 9.9.9.9
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Peer address change, add 57.1.1.6
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Sending label mapping msg
vc type 7, cbit 1, vc id 50, group id 6, vc label 21, status 0, mtu 1500
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Received label mapping msg, id 113
vc type 7, cbit 1, vc id 50, group id 6, vc label 21, status 0, mtu 1500
```

debug mpls l2transport vc

To display information about the status of the Any Transport over MPLS (AToM) virtual circuits (VCs), use the **debug mpls l2transport vc** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport vc {**event** | **fsm**}

no debug mpls l2transport vc {**event** | **fsm**}

Syntax Description

event	Displays AToM event messages about the VCs.
fsm	Displays the finite state machine.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

Usage Guidelines

You can issue this command from the line card or route processor.

Examples

The following is sample output from the **debug mpls l2transport vc** commands:

```
Router# debug mpls l2transport vc event

AToM vc event debugging is on

Router# debug mpls l2transport vc fsm

AToM vc fsm debugging is on

Router# show debugging

AToM:
  AToM vc event debugging is on
  AToM vc fsm debugging is on

*Mar 24 23:17:24.371: AToM MGR [9.9.9.9, 50]: Event provision, state changed from idle to
provisioned
*Mar 24 23:17:24.371: AToM MGR [9.9.9.9, 50]: Provision vc
*Mar 24 23:17:24.371: AToM SMGR [9.9.9.9, 50]: Requesting VC create, vc_handle 61A09930
*Mar 24 23:17:24.371: AToM MGR [9.9.9.9, 50]: Event local up, state changed from
provisioned to local standby
*Mar 24 23:17:24.371: AToM MGR [9.9.9.9, 50]: Update local vc label binding
*Mar 24 23:17:24.371: AToM SMGR [9.9.9.9, 50]: sucessfully processed create request
*Mar 24 23:17:24.875: %SYS-5-CONFIG_I: Configured from console by console
```

```
*Mar 24 23:17:25.131: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed
state to up

*Mar 24 23:17:28.567: AToM MGR [9.9.9.9, 50]: Event ldp up, state changed from local
standby to local ready
*Mar 24 23:17:28.567: AToM MGR [9.9.9.9, 50]: Advertise local vc label binding
*Mar 24 23:17:28.567: AToM MGR [9.9.9.9, 50]: Event remote up, state changed from local
ready to establishing
*Mar 24 23:17:28.567: AToM MGR [9.9.9.9, 50]: Remote end up
*Mar 24 23:17:28.567: AToM MGR [9.9.9.9, 50]: Event remote validated, state changed from
establishing to established
*Mar 24 23:17:28.567: AToM MGR [9.9.9.9, 50]: Validate vc, activating data plane
*Mar 24 23:17:28.567: AToM SMGR [9.9.9.9, 50]: Processing imposition update, vc_handle
61A09930, update_action 3, remote_vc_label 21
*Mar 24 23:17:28.567: AToM SMGR [9.9.9.9, 50]: Imposition Programmed, Output Interface:
PO5/0
*Mar 24 23:17:28.567: AToM SMGR [9.9.9.9, 50]: Processing disposition update, vc_handle
61A09930, update_action 3, local_vc_label 22
*Mar 24 23:17:28.571: AToM SMGR: Processing TFIB event for 9.9.9.9
*Mar 24 23:17:28.571: AToM SMGR [9.9.9.9, 50]: Imposition Programmed, Output Interface:
PO5/0
```

debug mpls ldp advertisements

To display information about the advertisement of labels and interface addresses to LDP peers, use the **debug mpls ldp advertisements** command in pivileged EXEC mode. To disable debugging output, use the **no** form of the command.

debug mpls ldp advertisements [**peer-acl** *acl*] [**prefix-acl** *acl*]

no debug mpls ldp advertisements [**peer-acl** *acl*] [**prefix-acl** *acl*]

Syntax Description

peer-acl *acl*	(Optional) Limits the displayed advertisements to those for LDP peers permitted by the access control list (ACL).
prefix-acl *acl*	(Optional) Limits the displayed advertisements to those for prefixes permitted by the ACL.

Defaults

Displays information about advertisements to all LDP peers for all prefixes.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

Use this command to monitor the label and address advertisements to LDP peers.

Use the **peer-acl** or **prefix-acl** options separately or together to limit the information display to specific LDP peers or specific prefixes.

Note This command monitors advertisement of non-LC-ATM labels (generic labels) only. Use the **debug mpls atm-ldp** command to monitor LC-ATM activity.

Examples

The following shows sample output from the **debug mpls ldp advertisements** command:

```
Router# debug mpls ldp advertisements

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 130.77.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 133.0.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 34.0.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 103.0.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 35.0.0.33
```

```
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 38.0.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 34.0.0.0/8, label 3 (#2)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 203.0.7.7/32, label 24 (#4)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 35.0.0.0/8, label 3 (#8)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 103.0.0.0/8, label 3 (#10)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 138.1.0.0/16, label 26 (#14)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 155.0.0.55/32, label 27 (#16)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 38.0.0.0/8, label 3 (#18)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 212.10.1.0/24, label 30 (#24)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 59.0.0.0/8, label 32 (#28)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 10.0.0.44/32, label 33 (#30)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 106.0.0.0/8, label 34 (#32)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 133.0.0.33/32, label 3 (#34)
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 45.0.0.0/8, label 39 (#36)
```

Table 164 describes the significant fields shown in the sample display above.

Table 164 debug mpls ldp advertisements Field Descriptions

Field	Description
tagcon:	Identifies the source of the message as the label control subsystem.
peer a.b.c.d:e	The LDP identifier of the peer to which the advertisement was targeted.
(pp 0xnnnnnnnn)	The identifier for the data structure used to represent the peer at the label distribution level. Useful for correlating debug output.
advertise X	Identifies what was advertised to the peer—either an interface address (“a.b.c.d”) or label binding (“a.b.c.d/m, label t (#n)”).
(#n)	For a label binding advertisement, the sequence number of the Label Information Base (LIB) modification that made it necessary to advertise the label.

Related Commands

Command	Description
show mpls ip binding	Displays label bindings known to the LSR.
debug mpls ldp bindings	Displays information about changes to the LIB used to keep track of label bindings learned from LDP peers through LDP downstream label distribution.
show mpls ldp neighbor	Displays the status of LDP sessions.

debug mpls ldp bindings

To display information about addresses and label bindings learned from LDP peers by means of LDP downstream unsolicited label distribution, use the **debug mpls ldp bindings** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command. use the **no** form of the command.

debug mpls ldp bindings [**peer-acl** *acl*] [**prefix-acl** *acl*]

no debug mpls ldp bindings [**peer-acl** *acl*] [**prefix-acl** *acl*]

Syntax Description

peer-acl *acl*	(Optional) Limits the displayed binding information to that learned from LDP peers permitted by the access control list (ACL).
prefix-acl *acl*	(Optional) Limits the displayed binding information to that learned for prefixes permitted by the ACL.

Defaults

Displays information about all bindings learned from all LDP peers.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

Use this command to monitor label bindings and LSR addresses learned from LDP peers.

Note This command monitors nonLC-ATM labels (generic labels) only. Use the **debug mpls atm-ldp** command to monitor LC-ATM activity.

Examples

The following shows sample output from the **debug mpls ldp bindings** command:

```
Router# debug mpls ldp bindings

tagcon:tibent(34.0.0.0/8):created; find route tags request
tagcon:tibent(34.0.0.0/8):label 3 (#2) assigned
tagcon:tibent(203.0.7.7/32):created; find route tags request
tagcon:tibent(203.0.7.7/32):label 24 (#4) assigned
tagcon:tibent(10.0.0.44/32):created; find route tags request
tagcon:tibent(10.0.0.44/32):label 33 (#30) assigned
tagcon:tibent(106.0.0.0/8):created; find route tags request
```

```
tagcon:tibent(106.0.0.0/8):label 34 (#32) assigned
tagcon:tibent(133.0.0.33/32):created; find route tags request
tagcon:tibent(133.0.0.33/32):label 3 (#34) assigned
tagcon:tibent(45.0.0.0/8):created; find route tags request
tagcon:tibent(45.0.0.0/8):label 39 (#36) assigned
tagcon:Assign peer id; 10.0.0.44:0:id 0
tagcon:10.0.0.44:0:10.0.0.44 added to addr<->ldp ident map
tagcon:10.0.0.44:0:34.0.0.44 added to addr<->ldp ident map
tagcon:10.0.0.44:0:45.0.0.44 added to addr<->ldp ident map
tagcon:tibent(10.0.0.44/32):rem label 3 from 10.0.0.44:0 added
tagcon:tibent(34.0.0.0/8):label 3 from 10.0.0.44:0 added
tagcon:tibent(45.0.0.0/8):label 3 from 10.0.0.44:0 added
tagcon:tibent(107.0.0.0/8):created; remote label learned
tagcon:tibent(107.0.0.0/8):label 55 from 10.0.0.44:0 added
tagcon:tibent(203.0.7.7/32):label 209 from 10.0.0.44:0 added
tagcon:tibent(133.0.0.33/32):label 207 from 10.0.0.44:0 added
```

Table 165 describes the significant fields shown in the sample display above.

Table 165 debug mpls ldp bindings Field Descriptions

Field	Description
tagcon:	Identifies the source of the message as the label control subsystem.
tibent(network/mask)	The destination that has a label binding change.
created; reason	A LIB entry has been created for the specified destination for the indicated reason.
rem label ...	Describes a change to the label bindings for the specified destination. The change is for a label binding learned from the specified LDP peer.
lcl label ...	Describes a change to a locally assigned (incoming) label for the specified destination.
(#n)	The sequence number of the modification to the LIB corresponding to the local label change.
a.b.c.d:n: e.f.g.h added to addr<->ldp ident map	The address e.f.g.h has been added to the set of addresses associated with LDP identifier a.b.c.d:n.

Related Commands

Command	Description
show mpls ldp bindings	Displays the contents of the Label Information Base (LIB).

debug mpls ldp messages

To display specific information (such as message type, source, and destination) regarding LDP messages sent to and received from LDP peers, use the **debug mpls ldp messages** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls ldp messages {**sent | received**} [**all**] [**peer-acl** *acl*]

no debug mpls ldp messages {**sent | received**} [**all**] [**peer-acl** *acl*]

Syntax Description

sent	Displays LDP messages sent to LDP peers permitted by the access control list (ACL).
received	Displays LDP messages received from LDP peers permitted by the ACL.
all	(Optional) Displays all LDP messages sent to and received from LDP peers (including periodic KeepAlive messages) permitted by the ACL.
peer-acl *acl*	(Optional) Limits the messages displayed for LDP peers in accordance with the ACL.

Defaults

All messages sent (for the **sent** keyword) or received (for the **received** keyword) are displayed except for periodic KeepAlive messages.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

LDP requires periodic transmission of KeepAlive messages. If you do not specify the **all** option, periodic KeepAlive messages are not displayed.

Examples

The following shows sample output from the **debug mpls ldp messages** command:

```
Router# debug mpls ldp messages received
Router# debug mpls ldp messages sent

ldp: Rcvd init msg from 10.0.0.44 (pp 0x0)
ldp: Sent init msg to 10.0.0.44:0 (pp 0x0)
ldp: Sent keepalive msg to 10.0.0.44:0 (pp 0x0)
ldp: Rcvd keepalive msg from 10.0.0.44:0 (pp 0x0)
ldp: Sent address msg to 10.0.0.44:0 (pp 0x610F00E0)
```

```
ldp: Sent label mapping msg to 10.0.0.44:0 (pp 0x610F00E0)
ldp: Sent label mapping msg to 10.0.0.44:0 (pp 0x610F00E0)
ldp: Sent label mapping msg to 10.0.0.44:0 (pp 0x610F00E0)
ldp: Rcvd address msg from 10.0.0.44:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 10.0.0.44:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 10.0.0.44:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 10.0.0.44:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 10.0.0.44:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 10.0.0.44:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 10.0.0.44:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 10.0.0.44:0 (pp 0x610F00E0)
```

Table 166 describes the significant fields shown in the sample display above.

Table 166 debug mpls ldp messages Field Descriptions

Field	Description
ldp:	Identifies the source of the displayed information as LDP.
Rcvd xxx msg Sent xxx msg	The type of message received or sent.
from a.b.c.d	The host that sent the message. Used in the early stages of the opening of an LDP session, when the LDP identifier is not yet known.
from a.b.c.d:e to a.b.c.d:e	The LDP identifier of the peer that sent the message or to which the message was sent.
(pp 0xnnnnnnnn)	Identifies the data structure used to represent the peer at the label distribution level. Useful for correlating debug output.

Related Commands

Command	Description
debug mpls ldp session io	Displays the contents of LDP messages sent to and received from LDP peers.

debug mpls ldp peer state-machine

To display information about state transitions for LDP sessions, use the **debug mpls ldp peer state-machine** command in privileged EXEC mode. To disable debugging output, use the **no** form of the command.

debug mpls ldp peer state-machine

no debug mpls ldp peer state-machine

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no options.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

LDP manages peer sessions by means of the following two coupled state machines:

- A low-level state machine that deals with session establishment and shutdown.
- A high-level state machine that deals with setting up and shutting down label advertisement

Use the **debug mpls ldp session state-machine** command to monitor the lower-level session state machine.

Use the **debug mpls ldp peer state-machine** command to monitor the higher-level session state machine.

Examples

The following shows sample output from the **debug mpls ldp peer state-machine** command:

```
Router# debug mpls ldp peer state-machine

tagcon: start session TCP timers for 10.0.0.44:0 (pp 0x610EEC84)
tagcon: Enqueue peer up work for 10.0.0.44:0 (pp 0x610EEC84)
tagcon: peer 10.0.0.44:0 (pp 0x610EEC84): Event unsol open
    unsol op pdg -> estab
tagcon: Send initial advertisements to peer 10.0.0.44:0
tagcon: Initial address advertisement to peer 10.0.0.44:0
tagcon: Initial label advertisement to peer 10.0.0.44:0
.
.
.
tagcon: peer 10.0.0.44:0 (pp 0x610EEC84): Event down
    estab -> destroyed
tagcon: peer 10.0.0.44:0 (pp 0x610EEC84): Event cleanup done
    destroyed -> non-ex
```

Table 167 describes the significant fields shown in the sample display.

Table 167 debug mpls ldp peer state-machine Field Descriptions

Field	Description
tagcon:	Identifies the source of the message as the label control subsystem.
a.b.c.d:e	The LDP identifier of the peer for the session with the state change.
(pp 0xnnnnnnnn)	Address of the data structure used to represent the peer at the label distribution level. This address is useful for correlating debug output.
Event E	The event causing the state change.
s1 -> s2	The state of the LDP session has changed from state s1 to state s2.

Related Commands

Command	Description
debug mpls ldp session io	Displays information about LDP messages sent to or received from LDP peers.
show mpls ldp neighbor	Displays the status of LDP sessions.

debug mpls ldp session io

To display the contents of LDP messages sent to and received from LDP peers, use the **debug mpls ldp session io** command in privileged EXEC mode. To disable debugging output, use the **no** form of the command.

debug mpls ldp session io [**all**] [**peer-acl** *acl*]

no debug mpls ldp session io [**all**] [**peer-acl** *acl*]

Syntax Description

all	(Optional) Includes the contents of periodic KeepAlive messages in the displayed message output to LDP peers.
peer-acl *acl*	(Optional) Limits the displayed message output to those LDP peers permitted by the access control list (ACL).

Defaults

Displays the contents of LDP messages sent and received except for periodic KeepAlive messages.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

Entering this command causes the contents of all messages sent and received, except for periodic KeepAlive messages, to be displayed.

Examples

The following shows sample output from the **debug mpls ldp session io** command:

```
Router# debug mpls ldp session io all

ldp: Rcvd init msg from 10.0.0.44 (pp 0x0)
ldp: LDP init msg: PDU hdr: LDP Id: 10.0.0.44:0; Msg Contents:
  0x00 0x01 0x00 0x20 0x90 0x00 0x00 0x2C 0x00 0x00 0x02 0x00 0x00 0x16 0x00 0x00
  0x10 0x21 0x05 0x00 0x00 0x0E 0x00 0x01 0x00 0xB4 0x00 0x00 0x00 0x00 0x85 0x00
  0x00 0x21 0x00 0x00
ldp: Sent init msg to 10.0.0.44:0 (pp 0x0)
ldp: LDP init msg: PDU hdr: LDP Id: 133.0.0.33:0; Msg Contents:
  0x00 0x01 0x00 0x20 0x85 0x00 0x00 0x21 0x00 0x00 0x02 0x00 0x00 0x16 0x00 0x00
  0x06 0x32 0x05 0x00 0x00 0x0E 0x00 0x01 0x00 0xB4 0x00 0x00 0x00 0x00 0x90 0x00
  0x00 0x2C 0x00 0x00
ldp: Sent keepalive msg to 10.0.0.44:0 (pp 0x0)
```

```
ldp: LDP keepalive msg: PDU hdr: LDP Id: 133.0.0.33:0; Msg Contents:
  0x00 0x01 0x00 0x0E 0x85 0x00 0x00 0x21 0x00 0x00 0x02 0x01 0x00 0x04 0x00 0x00
  0x06 0x33
ldp: Rcvd keepalive msg from 10.0.0.44:0 (pp 0x0)
ldp: LDP keepalive msg: PDU hdr: LDP Id: 10.0.0.44:0; Msg Contents:
  0x00 0x01 0x00 0x0E 0x90 0x00 0x00 0x2C 0x00 0x00 0x02 0x01 0x00 0x04 0x00 0x00
  0x10 0x22
ldp: Sent address msg to 10.0.0.44:0 (pp 0x610ECDD0)
ldp: LDP address msg: PDU hdr: LDP Id: 133.0.0.33:0; Msg Contents:
  0x00 0x01 0x00 0x34 0x85 0x00 0x00 0x21 0x00 0x00 0x03 0x00 0x00 0x2A 0x00 0x00
  0x06 0x34 0x01 0x01 0x00 0x22 0x00 0x01 0x02 0x00 0x00 0xA3 0x82 0x42 0x00 0x21
  0x82 0x4D 0x00 0x21 0x85 0x00 0x00 0x21 0x22 0x00 0x00 0x21 0x67 0x00 0x00 0x21
  0x23 0x00 0x00 0x21 0x26 0x00 0x00 0x21
ldp: Sent label mapping msg to 10.0.0.44:0 (pp 0x610ECDD0)
ldp: LDP label mapping msg: PDU hdr: LDP Id: 133.0.0.33:0; Msg Contents:
  0x00 0x01 0x00 0x22 0x85 0x00 0x00 0x21 0x00 0x00 0x04 0x00 0x00 0x18 0x00 0x00
  0x06 0x36 0x01 0x00 0x00 0x08 0x02 0x00 0x01 0x20 0xCB 0x00 0x07 0x07 0x02 0x00
  0x00 0x04 0x00 0x00 0x00 0x18
ldp: Rcvd address msg from 10.0.0.44:0 (pp 0x610ECDD0)
ldp: LDP address msg: PDU hdr: LDP Id: 10.0.0.44:0; Msg Contents:
  0x00 0x01 0x00 0x24 0x90 0x00 0x00 0x2C 0x00 0x00 0x03 0x00 0x00 0x1A 0x00 0x00
  0x10 0x23 0x01 0x01 0x00 0x12 0x00 0x01 0x90 0x00 0x00 0x2C 0x02 0x00 0x00 0xA4
  0x22 0x00 0x00 0x2C 0x2D 0x00 0x00 0x2C
ldp: Rcvd label mapping msg from 10.0.0.44:0 (pp 0x610ECDD0)
ldp: LDP label mapping msg: PDU hdr: LDP Id: 10.0.0.44:0; Msg Contents:
  0x00 0x01 0x00 0x22 0x90 0x00 0x00 0x2C 0x00 0x00 0x04 0x00 0x00 0x18 0x00 0x00
  0x10 0x24 0x01 0x00 0x00 0x08 0x02 0x00 0x01 0x20 0x90 0x00 0x00 0x2C 0x02 0x00
  0x00 0x04 0x00 0x00 0x00 0x03
```

Table 168 describes the significant fields shown in the sample display.

Table 168 debug mpls ldp session io Field Descriptions

Field	Description
ldp:	Identifies the source of the message as LDP.
Rcvd xxx msg	Indicates that a message of the specified type has been received.
from a.b.c.d	The host to which the message has been sent. Used in the early stages of the opening of an LDP session when the LDP identifier is not yet known.
Sent xxx msg	Indicates that a message of the specified type has been sent.
to a.b.c.d	The host to which the message has been sent. Used in the early stages of the opening of an LDP session when the LDP identifier is not yet known.
to a.b.c.d:e	The LDP identifier of the peer to which the message has been sent.
(pp 0xnnnnnnnn)	Identifies the data structure used to represent the peer at the label distribution level. Useful for correlating debug output.
--LDP xxx msg	The type of message that has been sent.
PDU hdr: LDP Id: a.b.c.d:e	The LDP identifier of the sender included in the LDP protocol data unit (PDU) header.
Msg Contents: 0xnn ... 0xnn	The contents of the message represented as a sequence of bytes.

debug mpls ldp session state-machine

To display information about state transitions for LDP sessions, use the **debug mpls ldp session state-machine** command in privileged EXEC mode. To disable this feature, use the **no** form of the command.

debug mpls ldp session state-machine [**peer-acl** *acl*]

no debug mpls ldp session state-machine [**peer-acl** *acl*]

Syntax Description

peer-acl *acl*	(Optional) Limits the displayed information to that for LDP peers permitted by the access control list (ACL).

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

LDP manages peer sessions by means of the following two coupled-state machines:

- A low-level state machine that deals with session establishment and shutdown
- A high-level state machine that deals with setting up and shutting down label advertisement

Use the **debug mpls ldp session state-machine** command to monitor the lower-level session state machine.

Use the **debug mpls ldp peer state-machine** command to monitor the higher-level session state machine.

Examples

The following shows sample output from the **debug mpls ldp session state-machine** command:

```
Router# debug mpls ldp session state-machine

ldp: ptcl_adj:10.0.0.44(0x610EED30): Non-existent -> Role pasv
ldp: create ptcl_adj: tp = 0x610EED30, ipaddr = 10.0.0.44
ldp: ptcl_adj:10.0.0.44(0x610EED30): Event: Xport opened;
    Role pasv -> Role pasv
ldp: ptcl_adj:34.0.0.44(0x610EED30): Event: Rcv Init;
    Role pasv -> Init rcvd pasv
ldp: ptcl_adj:34.0.0.44(0x610EED30): Event: Rcv KA;
    Init rcvd pasv -> Oper
ldp: ptcl_adj:unknown(0x610EED30): Event: Xport closed;
    Oper -> Non-existent
```

Table 169 describes the significant fields in the sample display.

Table 169 debug mpls tdp session state-machine Field Descriptions

Field	Description
ldp:	Identifies the source of the message as LDP.
ptcl_adj:a.b.c.d	Identifies the network address of the LDP peer.
(0xnnnnnnnn)	Identifies the data structure used to represent the peer at the protocol level. Useful for correlating debug output.
Event: E	The event that caused the state transition.
s1 -> s2	The state of the LDP session has changed from state s1 to state s2.

Related Commands

Command	Description
debug mpls ldp peer state-machine	Monitors the high-level peer session state machine.

debug mpls ldp targeted-neighbors

To display information about the target neighbor mechanism, use the **debug mpls ldp targeted-neighbors** command in privileged EXEC mode. To disable debugging output, use the **no** form of the command.

debug mpls ldp targeted-neighbors

no debug mpls ldp targeted-neighbors

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.

Usage Guidelines

Platforms that are not directly connected may engage in LDP label distribution (for example, to support two-level labeling across an LSP tunnel).

An LDP session between nondirectly connected LSRs is called a targeted session and is supported by LDP extended discovery which uses targeted Hello messages sent to specific IP addresses. This mechanism establishes LDP adjacencies to peers that are not directly adjacent, such as peers at either end of a tunnel.

An LSR (Router 1) attempting to initiate an LDP targeted session with another LSR (Router 2) sends targeted Hello messages sent to a specific IP address of Router 2. If the configuration of Router 2 permits it to respond to targeted Hello messages from Router 1, it does so, and the LDP session can be established. In this situation, Router 1 is said to be an active LSR for the targeted session because it initiated the targeted Hello messages; Router 2 is said to be a passive LSR for the session because it responded to them.

As with LDP sessions between two directly connected LSRs, it is possible for a targeted session to be the result of multiple discovery activities which are targeted to different IP addresses for the same LSR. In addition, it is possible for both LSRs in a targeted session to be active and for both to be passive.

The debug messages enabled by the **debug mpls ldp targeted-neighbors** command report activity relating to targeted sessions.

Examples

The following shows sample output from the **debug mpls ldp targeted-neighbors** command:

```
Router# debug mpls ldp targeted-neighbors

ldp-trgtnbr: 10.0.0.44 Req active
ldp-trgtnbr: 10.0.0.44 allocated
ldp-trgtnbr: 10.0.0.44 Set peer start; flags 0x0
ldp-trgtnbr: 10.0.0.44 Defer peer cleanup; cleancnt 1
ldp-trgtnbr: 10.0.0.44 Set peer finished; flags 0xF
ldp-trgtnbr: 10.0.0.44 ref count incremented to 1
ldp-trgtnbr: 10.0.0.44 Release active; ref count decremented to 0
ldp-trgtnbr: 10.0.0.44 Clear peer start; flags 0xF
ldp-trgtnbr: 10.0.0.44 Undefer cleanup start; clearcnt 0, flags 0xC
ldp-trgtnbr: 10.0.0.44 Undefer cleanup finish; clearcnt 0, flags 0x8
ldp-trgtnbr: 10.0.0.44 Clear peer finished; flags 0x8
ldp-trgtnbr: 10.0.0.44 freed
```

Table 170 describes the significant fields shown in the sample display.

Table 170 debug mpls ldp targeted-neighbors Field Descriptions

Field	Description
ldp-trgtnbr:	Identifies this as an LDP targeted neighbor debug statement.
10.0.0.44	IP address for the targeted neighbor.

Related Commands

Command	Description
show mpls ldp neighbor	Displays the status of LDP protocol sessions.