debug decnet adj through debug dss ipx event

debug decnet adj through debug dss ipx event

debug decnet adj

Note The **debugdecnetadj** command is not available in Cisco IOS Release 12.2(33)SXH and later Cisco IOS 12.2SX releases.

To display debugging information on DECnet adjacencies, use the **debugdecnetadj**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug decnet adj

no debug decnet adj

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Examples

The following is sample output from the **debugdecnetadj** command:

```
Router# debug decnet adj
DNET-ADJ: Level 1 hello from 1.3
DNET-ADJ: sending hellos
DNET-ADJ: Sending hellos to all routers on interface Ethernet0, blksize 1498
DNET-ADJ: Level 1 hello from 1.3
DNET-ADJ: 1.5 adjacency initializing
DNET-ADJ: sending triggered hellos
DNET-ADJ: Sending hellos to all routers on interface Ethernet0, blksize 1498
DNET-ADJ: Level 1 hello from 1.3
DNET-ADJ: 1.5 adjacency up
DNET-ADJ: Level 1 hello from 1.5
DNET-ADJ: 1.5 adjacency down, listener timeout
```

The following line indicates that the router is sending hello messages to all routers on this segment, which in this case is Ethernet 0:

```
DNET-ADJ: Sending hellos to all routers on interface Ethernet0, blksize 1498
```

The following line indicates that the router has heard a hello message from address 1.5 and is creating an adjacency entry in its table. The initial state of this adjacency will be *initializing*.

```
DNET-ADJ: 1.5 adjacency initializing
```

The following line indicates that the router is sending an unscheduled (triggered) hello message as a result of some event, such as new adjacency being heard:

```
DNET-ADJ: sending triggered hellos
```

The following line indicates that the adjacency with 1.5 is now up, or active:

```
DNET-ADJ: 1.5 adjacency up
```

The following line indicates that the adjacency with 1.5 has timed out, because no hello message has been heard from adjacency 1.5 in the time interval originally specified in the hello message from 1.5:

```
DNET-ADJ: 1.5 adjacency down, listener timeout
```

The following line indicates that the router is sending an unscheduled hello message, as a result of some event, such as the adjacency state changing:

```
DNET-ADJ: hello update triggered by state changed in dn_add_adjacency
```

debug decnet connects

The **debugdecnetconnects** command is not available in Cisco IOS Release 12.2(33)SXH and later Cisco IOS 12.2SX releases.

To display debugging information of all connect packets that are filtered (permitted or denied) by DECnet access lists, use the **debugdecnetconnects** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug decnet connects

no debug decnet connects

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Usage Guidelines

When you use connect packet filtering, it may be helpful to use the **decnetaccess-group** configuration command to apply the following basic access list:

```
access-list 300 permit 0.0 63.1023 eq any
```

You can then log all connect packets sent on interfaces to which you applied this list, in order to determine those elements on which your connect packets must be filtered.

Note Packet password and account information is not logged in the **debugdecnetconnects** message, nor is it displayed by the **showaccess** EXEC command. If you specify **password** or **account** information in your access list, they can be viewed by anyone with access to the configuration of the router.

Examples

The following is sample output from the**debugdecnetconnects** command:

```
Router# debug decnet connects
DNET-CON: list 300 item #2 matched src=19.403 dst=19.309 on Ethernet0: permitted
 srcname="RICK" srcuic=[0,017]
 dstobj=42 id="USER"
```

The table below describes significant fields shown in the output.

Table 1: debug decnet connects Field Descriptions

Field	Description
DNET-CON:	Indicates that this is a **debugdecnetconnects** packet.
list 300 item #2 matched	Indicates that a packet matched the second item in access list 300.
src=19.403	Indicates the source DECnet address for the packet.

Field	Description
dst=19.309	Indicates the destination DECnet address for the packet.
on Ethernet0:	Indicates the router interface on which the access list filtering the packet was applied.
permitted	Indicates that the access list permitted the packet.
srcname = "RICK"	Indicates the originator user of the packet.
srcuic=[0,017]	Indicates the source UIC of the packet.
dstobj=42	Indicates that DECnet object 42 is the destination.
id="USER"	Indicates the access user.

debug decnet events

Note The **debugdecnetevents** command is not available in Cisco IOS Release 12.2(33)SXH and later Cisco IOS 12.2SX releases.

To display debugging information on DECnet events, use the **debugdecnetevents**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug decnet events

no debug decnet events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Examples

The following is sample output from the**debugdecnetevents** command:

```
Router# debug decnet events
DNET: Hello from area 50 rejected - exceeded ‘max area' parameter (45)
DNET: Hello from area 50 rejected - exceeded ‘max area' parameter (45)
```

The following line indicates that the router received a hello message from a router whose area was greater than the max-area parameter with which this router was configured:

```
DNET: Hello from area 50 rejected - exceeded'max area' parameter (45)
```

The following line indicates that the router received a hello message from a router whose node ID was greater than the max-node parameter with which this router was configured:

```
DNET: Hello from node 1002 rejected - exceeded'max node' parameter (1000)
```

debug decnet packet

Note The **debugdecnetpacket**command is not available in Cisco IOS Release 12.2(33)SXH and later Cisco IOS 12.2SX releases.

To display debugging information on DECnet packet events, use the **debugdecnetpacket** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug decnet packet

no debug decnet packet

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Examples

The following is sample output from the **debugdecnetpacket** command:

```
Router# debug decnet packet
DNET-PKT: src 1.4 dst 1.5 sending to PHASEV
DNET-PKT: Packet fwded from 1.4 to 1.5, via 1.5, snpa 0000.3080.cf90, TokenRing0
```

The following line indicates that the router is sending a converted packet addressed to node 1.5 to Phase V:

```
DNET-PKT: src 1.4 dst 1.5 sending to PHASEV
```

The following line indicates that the router forwarded a packet from node 1.4 to node 1.5. The packet is being sent to the next hop of 1.5 whose subnetwork point of attachment (MAC address) on that interface is 0000.3080.cf90.

```
DNET-PKT: Packet fwded from 1.4 to 1.5, via 1.5, snpa 0000.3080.cf90, TokenRing0
```

debug decnet routing

Note The **debugdecnetrouting** command is not available in Cisco IOS Release 12.2(33)SXH and later Cisco IOS 12.2SX releases.

To display all DECnet routing-related events occurring at the router, use the **debugdecnetrouting** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug decnet routing

no debug decnet routing

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Examples

The following is sample output from the **debugdecnetrouting** command:

```
Router# debug decnet routing
DNET-RT: Received level 1 routing from 1.3 on Ethernet0 at 1:16:34
DNET-RT: Sending routes
DNET-RT: Sending normal routing updates on Ethernet0
DNET-RT: Sending level 1 routing updates on interface Ethernet0
DNET-RT: Level1 routes from 1.5 on Ethernet0: entry for node 5 created
DNET-RT: route update triggered by after split route pointers in dn_rt_input
DNET-RT: Received level 1 routing from 1.5 on Ethernet 0 at 1:18:35
DNET-RT: Sending L1 triggered routes
DNET-RT: Sending L1 triggered routing updates on Ethernet0
DNET-RT: removing route to node 5
```

The following line indicates that the router has received a level 1 update on Ethernet interface 0:

```
DNET-RT: Received level 1 routing from 1.3 on Ethernet0 at 1:16:34
```

The following line indicates that the router is sending its scheduled updates on Ethernet interface 0:

```
DNET-RT: Sending normal routing updates on Ethernet0
```

The following line indicates that the route will send an unscheduled update on this interface as a result of some event. In this case, the unscheduled update is a result of a new entry created in the routing table of the interface.

```
DNET-RT: route update triggered by after split route pointers in dn_rt_input
```

The following line indicates that the router sent the unscheduled update on Ethernet 0:

```
DNET-RT: Sending L1 triggered routes
DNET-RT: Sending L1 triggered routing updates on Ethernet0
```

The following line indicates that the router removed the entry for node 5 because the adjacency with node 5 timed out, or the route to node 5 through a next-hop router was disconnected:

```
DNET-RT: removing route to node 5
```

debug device-sensor

To enable debugging for device sensor, use the **debug device-sensor** command in privileged EXEC mode.

debug device-sensor {**errors** | **events**}

Syntax Description

errors	Displays device sensor error messages.
events	Displays messages for events such as protocol packet arrivals, identity updates, and release events sent to the session manager.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)SE1	This command was introduced.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.

Usage Guidelines

Use the **debug device-sensor** command in conjunction with the **debug authentication all** command to troubleshoot scenarios where device sensor cache entries are not being created for the connected devices.

Examples

The following is sample output from the **debug device-sensor events** command. The debug output shows how Cisco Discovery Protocol packets and Type-Length-Values (TLVs) are received from the device connected to the Gigabit Ethernet interface 2/1.

```
Device# debug device-sensor events

Device#
*Nov 30 23:58:45.811: DSensor: Received cdp packet from GigabitEthernet2/1:00d0.2bdf.08a5
*Nov 30 23:58:45.811: DSensor: SM returned no or invalid session label for
GigabitEthernet2/1:00d0.2bdf.08a5
*Nov 30 23:58:45.811: DSensor: Updating SM with identity attribute list
  cdp-tlv              0   00 01 00 0B 4A 41 45 30 37 34 31 31 50 53 32
  cdp-tlv              0   00 03 00 03 32 2F 38
  cdp-tlv              0   00 04 00 04 00 00 00 0A
  cdp-tlv              0   00 05 00 68 57 53 2D 43 32 39 34 38 20 53 6F 66 74 77 61 72 65
2C 20 56 65 72 73 69 6F 6E 20 4D 63 70 53 57 3A 20 36 2E 34 28 35 2E
 30 29 20 4E 6D 70 53 57 3A 20 36 2E 34 28 35 29 0A 43 6F 70 79 72 69 67 68 74 20 28 63 29
 20 31 39 39 35 2D 32 30 30 33 20 62 79 20 43 69 73 63 6F 20 53 79 73
74 65 6D 73 2C 20 49 6E 63 2E 0A
 cdp-tlv              0   00 06 00 08 57 53 2D 43 32 39 34 38
 cdp-tlv              0   00 09 00 00
 cdp-tlv              0   00 0A 00 02 00 21
 cdp-tlv              0   00 0B 00 01 01
 cdp-tlv              0   00 12 00 01 00
 cdp-tlv             0   00 13 00 01 00
 cdp-tlv             0   00 14 00 00
 cdp-tlv             0   00 15 00 0A 06 08 2B 06 01 04 01 09 05 2A
 cdp-tlv             0   00 16 00 16 00 00 00 02 01 01 CC 00 04 00 00 00 0001 01 CC 00 04
 01 01 01 01
 cdp-tlv             0   00 17 00 01 00
 swidb               0   604702240 (0x240B0620)
 clid-mac-addr       0   00 D0 2B DF 08 A5
*Nov 30 23:58:46.831: DSensor: Received cdp packet from GigabitEthernet2/1:00d0.2bdf.08a5exi
Switch#
*Nov 30 23:58:51.171: %SYS-5-CONFIG_I: Configured from console by console
```

Related Commands

Command	Description
debug authentication all	Displays all debugging information about the Authentication Manager and all features.
device-sensor accounting	Adds the device sensor protocol data to the accounting records and generates additional accounting events when new sensor data is detected.

debug dhcp

To display debugging information about the Dynamic Host Configuration Protocol (DHCP) client activities and to monitor the status of DHCP packets, use the **debugdhcp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dhcp [detail]

no debug dhcp [detail]

Syntax Description

detail	(Optional) Displays additional debugging information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.
12.3(8)T	The output of this command was enhanced to display default static routes.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

You can also use the **debugdhcp** command to monitor the subnet allocation and releasing for on-demand address pools.

For debugging purposes, the **debugdhcpdetail** command provides the most useful information such as the lease entry structure of the client and the state transitions of the lease entry. The debug output shows the scanned option values from received DHCP messages that are replies to a router request. The values of the op, htype, hlen, hops, server identifier option, xid, secs, flags, ciaddr, yiaddr, siaddr, and giaddr fields of the DHCP packet are shown in addition to the length of the options field.

Examples

The following examples show and explain some of the typical debugging messages you may see when using the **debugdhcpdetail** command.

The following sample output shows when a DHCP client sends a DHCPDISCOVER broadcast message to find its local DHCP server:

```
Router# debug dhcp detail
00:07:16:DHCP:DHCP client process started:10
00:07:16:RAC:Starting DHCP discover on Ethernet2
00:07:16:DHCP:Try 1 to acquire address for Ethernet2
00:07:16:%SYS-5-CONFIG_I:Configured from console by console
00:07:19:DHCP:Shutting down from get_netinfo()
00:07:19:DHCP:Attempting to shutdown DHCP Client
00:07:21:DHCP:allocate request
```

```
00:07:21:DHCP:new entry. add to queue
00:07:21:DHCP:SDiscover attempt # 1 for entry:
```

The first seven lines of the following output show the current values stored in the lease entry structure for the client:

```
00:07:21:Temp IP addr:0.0.0.0 for peer on Interface:Ethernet2
00:07:21:Temp sub net mask:0.0.0.0
00:07:21: DHCP Lease server:0.0.0.0, state:1 Selecting
00:07:21: DHCP transaction id:582
00:07:21: Lease:0 secs, Renewal:0 secs, Rebind:0 secs
00:07:21: Next timer fires after:00:00:03
00:07:21: Retry count:1   Client-ID:cisco-0010.7b6e.afd8-Et2
00:07:21:DHCP:SDiscover:sending 308 byte length DHCP packet
00:07:21:DHCP:SDiscover 308 bytes
00:07:21: B'cast on Ethernet2 interface from 0.0.0.0
```

The following output shows the offered addresses and parameters sent to the DHCP client by the DHCP server via a DHCPOFFER message. The messages containing the Scan field indicate the options that were scanned from the received BOOTP packet and the corresponding values:

```
00:07:23:DHCP:Received a BOOTREP pkt
00:07:23:DHCP:Scan:Message type:DHCP Offer
00:07:23:DHCP:Scan:Server ID Option:10.1.1.1 = A010101
00:07:23:DHCP:Scan:Lease Time:180
00:07:23:DHCP:Scan:Renewal time:90
00:07:23:DHCP:Scan:Rebind time:157
00:07:23:DHCP:Scan:Subnet Address Option:255.255.255.0
```

The following output shows selected fields in the received BOOTP packet:

```
00:07:23:DHCP:rcvd pkt source:10.1.1.1, destination: 255.255.255.255
00:07:23: UDP sport:43, dport:44, length:308
00:07:23: DHCP op:2, htype:1, hlen:6, hops:0
00:07:23: DHCP server identifier:10.1.1.1
00:07:23:   xid:582, secs:0, flags:8000
00:07:23:   client:0.0.0.0, your:10.1.1.2
00:07:23:   srvr: 0.0.0.0, gw:0.0.0.0
00:07:23:   options block length:60
00:07:23:DHCP Offer Message Offered Address:10.1.1.2
00:07:23:DHCP:Lease Seconds:180 Renewal secs: 90 Rebind secs:157
00:07:23:DHCP:Server ID Option:10.1.1.1
00:07:23:DHCP:offer received from 10.1.1.1
```

The following output shows when the DHCP client sends a DHCPREQUEST broadcast message to the DHCP server to accept the offered parameters:

```
00:07:23:DHCP:SRequest attempt # 1 for entry:
00:07:23:Temp IP addr:10.1.1.2 for peer on Interface:Ethernet2
00:07:23:Temp sub net mask:255.255.255.0
00:07:23:   DHCP Lease server:10.1.1.1, state:2 Requesting
00:07:23:   DHCP transaction id:582
00:07:23:   Lease:180 secs, Renewal:0 secs, Rebind:0 secs
00:07:23:   Next timer fires after:00:00:02
00:07:23:   Retry count:1   Client-ID:cisco-0010.7b6e.afd8-Et2
00:07:23:DHCP:SRequest- Server ID option:10.1.1.1
00:07:23:DHCP:SRequest- Requested IP addr option:10.1.1.2
00:07:23:DHCP:SRequest placed lease len option:180
00:07:23:DHCP:SRequest:326 bytes
00:07:23:DHCP:SRequest:326 bytes
00:07:23:  B'cast on Ethernet2 interface from 0.0.0.0
```

The following output shows when the DHCP server sends a DHCPACK message to the client with the full set of configuration parameters:

```
00:07:23:DHCP:Received a BOOTREP pkt
00:07:23:DHCP:Scan:Message type:DHCP Ack
00:07:23:DHCP:Scan:Server ID Option:10.1.1.1 = A010101
00:07:23:DHCP:Scan:Lease Time:180
00:07:23:DHCP:Scan:Renewal time:90
00:07:23:DHCP:Scan:Rebind time:157
```

```
00:07:23:DHCP:Scan:Subnet Address Option:255.255.255.0
00:07:23:DHCP:rcvd pkt source:10.1.1.1, destination: 255.255.255.255
00:07:23: UDP sport:43, dport:44, length:308
00:07:23: DHCP op:2, htype:1, hlen:6, hops:0
00:07:23: DHCP server identifier:10.1.1.1
00:07:23:   xid:582, secs:0, flags:8000
00:07:23:   client:0.0.0.0, your:10.1.1.2
00:07:23:   srvr: 0.0.0.0, gw:0.0.0.0
00:07:23:   options block length:60
00:07:23:DHCP Ack Message
00:07:23:DHCP:Lease Seconds:180 Renewal secs: 90 Rebind secs:157
00:07:23:DHCP:Server ID Option:10.1.1.1Interface Ethernet2 assigned DHCP address 10.1.1.2,
 mask 255.255.255.0
00:07:26:DHCP Client Pooling:***Allocated IP address:10.1.1.2
00:07:26:Allocated IP address = 10.1.1.2 255.255.255.0
```

The following output shows when a default gateway (option 3) is assigned a static IP address that is the default route and that static routes were added from the DHCP server:

```
*Oct 2 06:22:24: Setting default_gateway to 68.8.8.1 ! This is the option 3 default gateway.
*Oct 2 06:22:24: Adding default route 68.8.8.1
*Oct 2 06:22:24: DHCP: Adding static route to 4.3.2.1 255.255.255.255 via 68.8.8.1
*Oct 2 06:22:24: DHCP: Adding static route to 1.1.1.1 255.255.255.255 via 68.8.8.1
*Oct 2 06:22:24: DHCP: Adding static route to 67.2.2.2 255.255.255.255 via 68.8.8.1
```

Most fields are self-explanatory; however, fields that may need further explanation are described in the table below.

Table 2: debug dhcp Field Descriptions

Fields	Description
DHCP:Scan:Subnet Address Option:255.255.255.0	Subnet mask option (option 1).
DHCP server identifier:1.1.1.1	Value of the DHCP server ID option (option 54). Note that this is not the same as the siaddr field, which is the server IP address.
srvr:0.0.0.0, gw:0.0.0.0	srvr is the value of the siaddr field. gw is the value of the giaddr field.

Related Commands

Command	Description
debug ip ddns update	Enables debugging for DDNS updates.
debug ip dhcp server	Enables DHCP server debugging.
host (host-list)	Specifies a list of hosts that will receive DDNS updates of A and PTR RRs.
ip ddns update hostname	Enables a host to be used for DDNS updates of A and PTR RRs.
ip ddns update method	Specifies a method of DDNS updates of A and PTR RRs and the maximum interval between the updates.

Command	Description
ip dhcp client update dns	Enables DDNS updates of A RRs using the same hostname passed in the hostname and FQDN options by a client.
ip dhcp-client update dns	Enables DDNS updates of A RRs using the same hostname passed in the hostname and FQDN options by a client.
ip dhcp update dns	Enables DDNS updates of A and PTR RRs for most address pools.
ip host-list	Specifies a list of hosts that will receive DDNS updates of A and PTR RRs.
show ip ddns update	Displays information about the DDNS updates.
show ip ddns update method	Displays information about the DDNS update method.
show ip dhcp server pool	Displays DHCP server pool statistics.
show ip host-list	Displays the assigned hosts in a list.
update dns	Dynamically updates a DNS with A and PTR RRs for some address pools.

debug dhcp redundancy

To display debugging information about DHCP proxy client redundancy events, use the **debugdhcpredundancy**command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug dhcp redundancy

no debug dhcp redundancy

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging output is disabled for DHCP redundancy events.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(31)SB2	This command was introduced.
12.2(31)SRB1	This command was integrated into Cisco IOS Release 12.2(31)SRB1.

Examples

The following example displays debug messages regarding DHCP redundancy events. The last line is output when the **debugdhcpredundancy** command is enabled. The line indicates that the active Route Processor has sent a dynamic lease synchronization message for IP address 10.1.1.1:

```
Router# debug dhcp redundancy
*Mar 15 10:32:21: DHCPD: assigned IP address 10.1.1.1 to client
*Mar 15 10:32:21: DHCPD: dynamic sync sent for 10.1.1.1
```

Related Commands

Command	Description
debug ip dhcp server redundancy	Displays debugging information about DHCP server and relay agent redundancy events.

debug dialer events

To display debugging information about the packets received on a dialer interface, use the**debugdialerevents** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dialer events

no debug dialer events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

When dial-on-demand routing (DDR) is enabled on the interface, information concerning the cause of any call (called the *Dialingcause*) is displayed.

Examples

In the following example, the line of output for an IP packet lists the name of the DDR interface and the source and destination addresses of the packet:

```
Router# debug dialer events
Dialing cause: Serial0: ip (s=172.16.1.111 d=172.16.2.22)
```

The following line of output for a bridged packet lists the DDR interface and the type of packet (in hexadecimal). For information on these packet types, see the "Ethernet Type Codes" appendix of the Cisco IOS Bridging and IBM Networking Command Reference publication.

```
Dialing cause: Serial1: Bridge (0x6005)
```

Most messages are self-explanatory; however, messages that may need some explanation are described in the table below.

Table 3: debug dialer events Message Descriptions

Message	Description
Dialer0: Already *xxx* call(s) in progress on Dialer0, dialing not allowed	Number of calls in progress (*xxx*) exceeds the maximum number of calls set on the interface.
Dialer0: No free dialer - starting fast idle timer	All the lines in the interface or rotary group are busy, and a packet is waiting to be sent to the destination.
BRI0: rotary group to *xxx* overloaded (*yyy*)	Number dialer (*xxx*) exceeds the load set on the interface (*yyy*).
BRI0: authenticated host *xxx* with no matching dialer profile	No dialer profile matches *xxx*, the Challenge Handshake Authentication Protocol (CHAP) name or remote name of the remote host.

Message	Description
BRI0: authenticated host *xxx* with no matching dialer map	No dialer map matches *xxx*, the CHAP name or remote name of the remote host.
BRI0: Can’t place call, verify configuration	Dialer string or dialer pool on an interface not set.

The table below describes the messages that the **debugdialerevents**command can generate for a serial interface used as a V.25*bis* dialer for DDR.

Table 4: debug dialer events Command Message Descriptions for DDR

Message	Description
Serial 0: Dialer result = xxxxxxxxxx	Result returned from the V.25*bis*dialer. It is useful in debugging if calls are failing. On some hardware platforms, this message cannot be displayed due to hardware limitations. Possible values for the xxxxxxxxxx variable depend on the V.25*bis* device with which the router is communicating.
Serial 0: No dialer string defined. Dialing cannot occur.	Packet is received that should cause a call to be placed. However, no dialer string is configured, so dialing cannot occur. This message usually indicates a configuration problem.
Serial 0: Attempting to dial xxxxxxxxxx	Packet has been received that passes the dial-on-demand access lists. That packet causes phone number xxxxxxxxxx to be dialed.
Serial 0: Unable to dial xxxxxxxxxx	Phone call to xxxxxxxxxx cannot be placed. This failure might be due to a lack of memory, full output queues, or other problems.
Serial 0: disconnecting call	Router hangs up a call.
Serial 0: idle timeout Serial 0: re-enable timeout Serial 0: wait for carrier timeout	One of these three messages is displayed when a dialer timer expires. These messages are mostly informational, but are useful for debugging a disconnected call or call failure.

Related Commands

Command	Description
debug decnet packet	Displays debugging information about the packets received on a dialer interface.

debug dialer forwarding

To display debugging information about the control plane at the home gateway (HGW) for Layer 2 Tunneling Protocol (L2TP) dialout, use the **debugdialerforwarding**command in privileged EXEC mode. The**no** form of this command disables debugging output.

debug dialer forwarding

no debug dialer forwarding

Syntax Description

This command has no keywords or arguments.

Command Default

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2 T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use the **debugdialerforwarding** command to configure a virtual private dialout network (VPDN) on the HGW and a network access server (NAS) to dial from the HGW to the client.

An L2TP tunnel is created between the HGW and the NAS and the packets are forwarded transparently at the NAS.

Examples

The following is sample output from the **debugdialerforwarding**command for dialing from the HGW to the client.

Note DDR-FWD is **debugdialerforwarding** information. (DDR= dial-on-demand routing.)

```
Router# debug dialer forwarding
Dialer forwarding events debugging is on
Router# ping
Protocol [ip]:
Target IP address:1.1.1.3
Repeat count [5]:1
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 1.1.1.3, timeout is 2 seconds:
```

```
1d00h:Vi3 DDR-FWD 83093A60:event [REQUEST]  state before [IDLE]
1d00h:Vi3 DDR-FWD 83093A60:VPN Authorization started
1d00h:Vi3 DDR-FWD 83093A60:VPN author result 1
1d00h:Vi3 DDR-FWD 83093A60:event [AUTHOR FOUND]  state before [AUTHORIZING]
1d00h:Vi3 DDR-FWD 83093A60:event [FORWARDED]  state before [FORWARDING]
1d00h:Vi3 DDR-FWD 83093A60:Connection is up, start LCP now
*Mar  2 00:31:33:%LINK-3-UPDOWN:Interface Virtual-Access3, changed state to up.
Success rate is 0 percent (0/1)
R2604#
*Mar  2 00:31:35:%LINEPROTO-5-UPDOWN:Line protocol on Interface Virtual-Access3, changed
state to up
Router#
```

Outgoing call disconnected:

```
Router#
1d00h:Vi3 DDR-FWD 83093A60:event [VPDN DISC]  state before [FORWARDED]
*Mar  2 00:33:33:%LINK-3-UPDOWN:Interface Virtual-Access3, changed state to down
*Mar  2 00:33:34:%LINEPROTO-5-UPDOWN:Line protocol on Interface Virtual-Access3, changed
state to down
```

Related Commands

Command	Description
debug dialer events	Displays debugging information about events on a dialer interface.
debug dialer packets	Displays debugging information about packets received on a dialer interface.

debug dialer map

To display debugging information about the creation and deletion of dynamic dialer maps, use the **debugdialermap** command in privileged EXEC mode. The**no** form of this command disables debugging output.

debug dialer map

no debug dialer map

Syntax Description

This command has no keywords or arguments.

Command Default

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5.1)	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use the **debugdialermap** command to track large-scale dialout (LSDO) and incoming calls that use dynamic dialer maps. This command shows the whole trace including when the map is created and removed.

If an interface is configured for dial-on-demand routing (DDR), and a map to a specified address does not exist, then a dynamic dialer map is created and when the call disconnects, the dialer map is removed.

Note Do not configure a dialer string or a dialer map on the incoming interface.

Examples

In the following sample output from the **debugdialermap** command, a dialer map is created when an incoming call is connected and removed when that call is disconnected:

```
Router# debug dialer map
Dial on demand dynamic dialer maps debugging is on
```

Incoming call connected:

```
Router#
*Mar 22 12:19:15.597:%LINK-3-UPDOWN:Interface BRI0/0:1, changed state to up
*Mar 22 12:19:17.748:BR0/0:1 DDR:dialer_create_dynamic_map map created for 11.0.0.1
*Mar 22 12:19:18.734:%LINEPROTO-5-UPDOWN:Line protocol on Interface BRI0/0:1, changed state
 to up
*Mar 22 12:19:21.598:%ISDN-6-CONNECT:Interface BRI0/0:1 is now connected to unknown R2604
```

Incoming call disconnected:

```
Router#
*Mar 22 12:21:15.597:%ISDN-6-DISCONNECT:Interface BRI0/0:1  disconnected from  R2604, call
 lasted 120 seconds
*Mar 22 12:21:15.645:%LINK-3-UPDOWN:Interface BRI0/0:1, changed state to down
*Mar 22 12:21:15.649:BR0/0:1 DDR:dialer_remove_dynamic_map map 11.0.0.1 removed
*Mar 22 12:21:16.647:%LINEPROTO-5-UPDOWN:Line protocol on Interface BRI0/0:1, changed state
 to down
```

Related Commands

Command	Description
debug dialer events	Displays debugging information about events on a dialer interface.
debug dialer packets	Displays debugging information about packets received on a dialer interface.

debug dialpeer

Note Effective with release 12.3(8)T, the **debugdialpeer** command is replaced by the **debugvoipdialpeer** command. See the **debugvoipdialpeer** command for more information.

To view dial peer information, use the **debugdialpeer**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dialpeer

no debug dialpeer

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced.
12.3(8)T	This command was replaced by the **debugvoipdialpeer** command.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug**dialpeer command. Using the **debug**dialpeer command generates a large volume of debugging messages, which can affect router performance.

Examples

The following is sample output for the **debug**dialpeer command. The output shows the destination pattern configured on the matched dial-peer. Expanded string is the string after applying number translation to the original number. It shows that dial-peer 1311 was an incoming dial-peer match. It also shows that routing label was att1. It shows that dial-peer 5108888 and 111399 are an outgoing dial-peer match.

```
Router# debug dialpeer
Router#
00:22:28: Inside dpMatchCore:
00:22:28: destination pattn:5108880101 expanded string:5108880101
00:22:28:MatchNextPeer:Peer 1311 matched
00:22:28: Inside dpMatchCore:
00:22:28: destination pattn:5108880101 expanded string:5108880101
00:22:28: Inside dpMatchCore:
00:22:28: destination pattn:4088880101 expanded string:4088880101
00:22:28: Inside dpMatchCore:
00:22:28: destination pattn:4088880101 expanded string:4088880101
00:22:28: dpAssociateIncomingPeer_T:Matching route label
att1
```

```
00:22:28: Inside dpMatchCore:
00:22:28: destination pattn:5108880101 expanded string:5108880101
00:22:28: dpAssociateIncomingPeer_T:Matching peer with src route label att1 failed
00:22:28: Inside dpMatchCore:
00:22:28: destination pattn:5108880101 expanded string:5108880101
00:22:28:MatchNextPeer:Peer 1311 matched
00:22:28: Inside dpMatchPeersMoreArg
00:22:28:dpMatchPeersMoreArg:Match Dest. pattern; called (5108880101)
00:22:28: Inside dpMatchCore:
00:22:28: destination pa
Router#ttn:5108880101 expanded string:5108880101
00:22:28:MatchNextPeer:Peer 5108888 matched
00:22:28:MatchNextPeer:Peer 111399 matched
00:22:28:dpMatchPeersMoreArg:Result=0 after MATCH_ORIGINATE
```

The table below describes the significant fields shown in the display.

Table 5: debug dialpeer Field Descriptions

Field	Description
destination pattn	Destination pattern configured on the dial peer.
expanded string	The string after applying number translation to the original number.
Match Dest. pattern; called	Indicates that dial-peer match is going to match destination pattern against the called number.
Matching route label	The trunk group label or carrier id that is used for matching a dial peer.
MatchNextPeer	Indicates the dial peer tag that matched.
Result	Indicates the result of dial peer matching algorithm: 0 = Successful 1 = More digits needed for a possible match -1 = No match (match failed) -2 = The digits matched, but the destination address could not be obtained

Related Commands

Command	Description
call-block (dial peer)	Enables blocking of incoming calls on the dial peer.
carrier-id (dial-peer)	Identifies the carrier handling the incoming call.
session target (ENUM)	Specifies the ENUM search table for the target session.
show dial-peer voice	Displays the configuration of the dial peer.
translation-profile (dial-peer)	Assigns a translation profile to the dial peer.

Command	Description
trunkgroup (dial-peer)	Assigns a trunk group to the dial peer.
trunk-group-label (dial-peer)	Identifies the trunk group handling the incoming call.

debug diameter

To display information about the Diameter Protocol, use the **debugdiameter** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug diameter [**dcca**| **connection**| **error**| **packet**| **event**| **fsm**| **failover**]

no debug diameter [**dcca**| **connection**| **error**| **packet**| **event**| **fsm**| **failover**]

Syntax Description

dcca	(Optional) Enables debugging for Diameter-Credit Control Accounting.
connection	(Optional) Enables debugging output for the connection between two Diameter nodes.
error	(Optional) Enables debugging output for Diameter errors.
packet	(Optional) Enables debugging output for Diameter data packets.
event	(Optional) Enables debugging output for Diameter events.
fsm	(Optional) Enables debugging output for the finite state machine.
failover	(Optional) Enables debugging output for Diameter redundancy.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to display information about any of the listed classes of information about the Diameter Protocol.

Examples

The following examples show output from the **debugdiameter** command:

Examples

```
Router# debug diameter all
*May  9 17:58:14.832: Dia Base: Diameter Peer configured. Allocate connection context.
*May  9 17:58:14.832: Dia Base: Allocate the peer connection context 50F63888, handle
C000000C *May  9 17:58:14.832: Dia Base: (C000000C): Received peer configuration event *May
  9 17:58:14.832: Dia Peer FSM (50F63888): input event START in state CLOSED *May  9
17:58:14.832: Dia Peer FSM (50F63888): Starting Connection timer *May  9 17:58:14.832: Dia
 Peer FSM (50F63888): event START, state
CLOSED-->WAIT_CONN_ACK
*May  9 17:58:14.836: Dia Transport: socket 0 - connecting to 9.113.33.6
(3868)
*May  9 17:58:14.836: Dia Transport: socket 0 - connection in progress *May  9 17:58:14.836:
 Dia Transport: socket 0 - local address 9.113.33.5
(49214)
*May  9 17:58:14.836: Dia Transport: socket 0 - resume socket write - nothing to write *May
  9 17:58:14.836: Dia Base: (C000000C): Received peer connection event from transport *May
  9 17:58:14.836: Dia Peer FSM (50F63888): input event RCV_CONN_ACK in state WAIT_CONN_ACK
 *May  9 17:58:14.836: Dia Base: Sending diameter message to peer "Unknown"
*May  9 17:58:14.836: DIAMETER:  CER message, ver=1, len=120, app=0, [2328318322/2328318322]

*May  9 17:58:14.836: DIAMETER:  Origin-host-name              [264]
"host"                         (M)
*May  9 17:58:14.836: DIAMETER:  Origin-Realm                  [296]
"cisco"                        (M)
*May  9 17:58:14.836: DIAMETER:  Host-IP-address               [257]
9.113.33.5                     (M)
*May  9 17:58:14.836: DIAMETER:  Vendor-ID                     [266]   9
(M)
*May  9 17:58:14.836: DIAMETER:  Product-name                  [269]
"C7200-G8IS-M"
*May  9 17:58:14.836: DIAMETER:  Auth-Application-ID           [258]   4
(M)
*May  9 17:58:14.836: DIAMETER:  Firmware-Revision             [267]   1
50D0B710:          01000078 80000101 00000000      ...x........
50D0B720: 8AC75172 8AC75172 00000108 4000000C  .GQr.GQr....@...
50D0B730: 686F7374 00000128 4000000D 63697363  host...(@...cisc
50D0B740: 6F000000 00000101 4000000E 00010971  o.......@......q
50D0B750: 21050000 0000010A 4000000C 00000009  !.......@.......
50D0B760: 0000010D 00000014 43373230 302D4738  ........C7200-G8
50D0B770: 49532D4D 00000102 4000000C 00000004  IS-M....@.......
50D0B780: 0000010B 0000000C 00000001 00        .............
*May  9 17:58:14.836: Dia Base: Request message hash ctx created for [2328318322/2328318322]
 *May  9 17:58:14.836: Dia Peer FSM (50F63888): Starting CER timer *May  9 17:58:14.836:
Dia Peer FSM (50F63888): event RCV_CONN_ACK, state WAIT_CONN_ACK-->WAIT_CEA *May  9
17:58:14.836: Dia Transport: Dia Transport write message event *May  9 17:58:14.836: Dia
Transport: socket 0 - complete msg sent *May  9 17:58:14.840: Dia Transport: socket 0 -
complete read of 20 bytes *May  9 17:58:14.840: Dia Transport: complete header read from
socket 0 *May  9 17:58:14.840: Dia Transport: read msg (172) bytes from socket 0 *May  9
17:58:14.840: Dia Transport: socket 0 - complete read of 172 bytes *May  9 17:58:14.840:
Dia Base: Diameter message received from the peer "Unknown"
*May  9 17:58:14.840: DIAMETER:  CEA message, ver=1, len=192, app=0, [2328318322/2328318322]

*May  9 17:58:14.840: DIAMETER:  Result-code                   [268]
2001                           (M)
*May  9 17:58:14.840: DIAMETER:  Origin-host-name              [264]
"diameter2.cisco.com"          (M)
*May  9 17:58:14.840: DIAMETER:  Origin-Realm                  [296]
"cisco.com"                    (M)
*May  9 17:58:14.840: DIAMETER:  Host-IP-address               [257]
10.77.154.80                   (M)
*May  9 17:58:14.840: DIAMETER:  Vendor-ID                     [266]   9
(M)
*May  9 17:58:14.840: DIAMETER:  Product-name                  [269]
"Diameter-Server"
*May  9 17:58:14.840: DIAMETER:  Supported-Vendor-ID           [265]
10415                          (M)
```

```
*May  9 17:58:14.840: DIAMETER:  Supported-Vendor-ID           [265]
12645                          (M)
*May  9 17:58:14.840: DIAMETER:  Supported-Vendor-ID           [265]   9
(M)
*May  9 17:58:14.840: DIAMETER:  Supported-Vendor-ID           [265]   9
(M)
*May  9 17:58:14.840: DIAMETER:  Auth-Application-ID           [258]   4
(M)
65940780:          010000C0 00000101 00000000      ...@........
65940790: 8AC75172 8AC75172 0000010C 4000000C  .GQr.GQr....@...
659407A0: 000007D1 00000108 4000001B 6469616D  ...Q....@...diam
659407B0: 65746572 322E6369 73636F2E 636F6D00  eter2.cisco.com.
659407C0: 00000128 40000011 63697363 6F2E636F  ...(@...cisco.co
659407D0: 6D000000 00000101 4000000E 00010A4D  m.......@......M
659407E0: 9A500000 0000010A 4000000C 00000009  .P......@.......
659407F0: 0000010D 00000017 4469616D 65746572  ........Diameter
65940800: 2D536572 76657200 00000109 4000000C  -Server.....@...
65940810: 000028AF 00000109 4000000C 00003165  ..(/....@.....1e
65940820: 00000109 4000000C 00000009 00000109  ....@...........
65940830: 4000000C 00000009 00000102 4000000C  @...........@...
65940840: 00000004 00                          .....
*May  9 17:58:14.840: Dia Base: Request message hash ctx removed for [2328318322/2328318322]
 *May  9 17:58:14.840: Dia Base: (C000000C): Received msg event from message i/o *May  9
17:58:14.840: Dia Peer FSM (50F63888): input event RCV_CEA in state WAIT_CEA *May  9
17:58:14.840: Dia Peer FSM (50F63888): Starting Watchdog timer *May  9 17:58:14.840:
%DIABASE-4-DIA_PEER_UP: Diameter peer 9.113.33.6 port 3868 TCP UP *May  9 17:58:14.840: Dia
 Peer FSM (50F63888): event RCV_CEA, state WAIT_CEA-->OPEN
```

Examples

```
*May  9 17:59:14.840: Dia Peer FSM (50F63888): input event TIMEOUT in
state OPEN
*May  9 17:59:14.840: Dia Base: Sending diameter message to peer
"diameter2.cisco.com"
*May  9 17:59:14.840: DIAMETER:  DWR message, ver=1, len=48, app=0,
[2328318323/2328318323]
*May  9 17:59:14.840: DIAMETER:  Origin-host-name              [264]
"host"                         (M)
*May  9 17:59:14.840: DIAMETER:  Origin-Realm                  [296]
"cisco"                        (M)
50D0B710:          01000030 80000118 00000000      ...0........
50D0B720: 8AC75173 8AC75173 00000108 4000000C  .GQs.GQs....@...
50D0B730: 686F7374 00000128 4000000D 63697363  host...(@...cisc
50D0B740: 6F000000 FD                          o...}
*May  9 17:59:14.840: Dia Base: Request message hash ctx created for
[2328318323/2328318323]
*May  9 17:59:14.840: Dia Peer FSM (50F63888): Starting Watchdog timer,
[60] left for next timeout*May  9 17:59:14.840: Dia Peer FSM (50F63888):
event TIMEOUT, state OPEN-->OPEN
*May  9 17:59:14.840: Dia Transport: Dia Transport write message event
*May  9 17:59:14.840: Dia Transport: socket 0 - complete msg sent
*May  9 17:59:14.840: Dia Transport: socket 0 - complete read of 20
bytes
*May  9 17:59:14.840: Dia Transport: complete header read from socket 0
*May  9 17:59:14.840: Dia Transport: read msg (60) bytes from socket 0
*May  9 17:59:14.840: Dia Transport: socket 0 - complete read of 60
bytes
*May  9 17:59:14.840: Dia Base: Diameter message received from the peer
"diameter2.cisco.com"
*May  9 17:59:14.840: DIAMETER:  DWA message, ver=1, len=80, app=0,
[2328318323/2328318323]
*May  9 17:59:14.840: DIAMETER:  Result-code                   [268]
2001                           (M)
*May  9 17:59:14.840: DIAMETER:  Origin-host-name              [264]
"diameter2.cisco.com"          (M)
*May  9 17:59:14.840: DIAMETER:  Origin-Realm                  [296]
"cisco.com"                    (M)
65940780:          01000050 00000118 00000000      ...P........
65940790: 8AC75173 8AC75173 0000010C 4000000C  .GQs.GQs....@...
659407A0: 000007D1 00000108 4000001B 6469616D  ...Q....@...diam
659407B0: 65746572 322E6369 73636F2E 636F6D00  eter2.cisco.com.
659407C0: 00000128 40000011 63697363 6F2E636F  ...(@...cisco.co
659407D0: 6D000000 00                          m....
*May  9 17:59:14.840: Dia Base: Request message hash ctx removed for
[2328318323/2328318323]
*May  9 17:59:14.840: Dia Base: (C000000C): Received msg event from
message i/o
*May  9 17:59:14.840: Dia Peer FSM (50F63888): input event RCV_DWA in
state OPEN
*May  9 17:59:14.840: Dia Peer FSM (50F63888): Starting Watchdog timer
*May  9 17:59:14.840: Dia Peer FSM (50F63888): event RCV_DWA, state
OPEN-->OPEN
```

Examples

```
*May  9 18:07:18.472: Dia Transport: socket 0 READ event: UP->CLOSE due
to bytes read = 0
*May  9 18:07:18.472: Dia Base: (8600000E): Received peer disconnection
event from transport
*May  9 18:07:18.472: %DIABASE-4-DIA_PEER_DOWN: Diameter peer 9.113.33.6
port 3868 TCP DOWN
*May  9 18:07:18.472: Dia Peer FSM (2068FF44): input event PEER_DISC in
state OPEN
*May  9 18:07:18.472: Dia Peer FSM (2068FF44): Starting Reconnect timer
*May  9 18:07:18.472: Dia Peer FSM (2068FF44): event PEER_DISC, state
OPEN-->CLOSED
*May  9 18:07:48.472: Dia Peer FSM (2068FF44): input event START in
state CLOSED
*May  9 18:07:48.472: Dia Peer FSM (2068FF44): Starting Connection timer
*May  9 18:07:48.472: Dia Peer FSM (2068FF44): event START, state
CLOSED-->WAIT_CONN_ACK
*May  9 18:07:48.472: Dia Transport: socket 0 - connecting to 9.113.33.6
(3868)
*May  9 18:07:48.472: Dia Transport: socket 0 - connection in progress
*May  9 18:07:48.472: Dia Transport: socket 0 - local address 9.113.33.5
(61122)
*May  9 18:07:48.472: Dia Transport: socket 0 - CONN_WAIT->CLOSE
*May  9 18:07:48.472: Dia Base: (8600000E): Received peer disconnection
event from transport
*May  9 18:07:48.472: Dia Peer FSM (2068FF44): input event PEER_DISC in
state WAIT_CONN_ACK
*May  9 18:07:48.472: Dia Peer FSM (2068FF44): Starting Reconnect timer
*May  9 18:07:48.472: Dia Peer FSM (2068FF44): event PEER_DISC, state
WAIT_CONN_ACK-->CLOSED
```

Examples

```
Ginger(config)#no diameter peer watch
Ginger(config)#
*May  9 18:05:02.812: Dia Base: Peer unconfigured, start peer
disconnection
*May  9 18:05:02.812: Dia Base: (C000000C): Received peer
unconfiguration event
*May  9 18:05:02.812: Dia Peer FSM (50F63888): input event STOP in state
OPEN
*May  9 18:05:02.812: Dia Base: Sending diameter message to peer
"diameter2.cisco.com"
*May  9 18:05:02.812: DIAMETER:  DPR message, ver=1, len=60, app=0,
[2328318329/2328318329]
*May  9 18:05:02.812: DIAMETER:  Origin-host-name              [264]
"host"                         (M)
*May  9 18:05:02.816: DIAMETER:  Origin-Realm                  [296]
"cisco"                        (M)
*May  9 18:05:02.816: DIAMETER:  Peer-disconnect-reason        [273]
Server-do-not-want-to-talk     (M)
653D1810:                   0100003C 8000011A          ...<....
653D1820: 00000000 8AC75179 8AC75179 00000108  .....GQy.GQy....
653D1830: 4000000C 686F7374 00000128 4000000D  @...host...(@...
653D1840: 63697363 6F000000 00000111 4000000C  cisco.......@...
653D1850: 00000002 00                          .....
*May  9 18:05:02.816: Dia Base: Request message hash ctx created for
[2328318329/2328318329]
*May  9 18:05:02.816: Dia Peer FSM (50F63888): Starting DPR timer
```

```
*May  9 18:05:02.816: Dia Peer FSM (50F63888): event STOP, state
OPEN-->CLOSING
*May  9 18:05:02.816: Dia Transport: Dia Transport write message event
*May  9 18:05:02.816: Dia Transport: socket 0 - complete msg sent
*May  9 18:05:02.816: Dia Transport: socket 0 - complete read of 20
bytes
*May  9 18:05:02.816: Dia Transport: complete header read from socket 0
*May  9 18:05:02.816: Dia Transport: read msg (60) bytes from socket 0
*May  9 18:05:02.816: Dia Transport: socket 0 - complete read of 60
bytes
*May  9 18:05:02.816: Dia Base: Diameter message received from the peer
"diameter2.cisco.com"
*May  9 18:05:02.816: DIAMETER:  DPA message, ver=1, len=80, app=0,
[2328318329/2328318329]
*May  9 18:05:02.816: DIAMETER:  Result-code                   [268]
2001                           (M)
*May  9 18:05:02.816: DIAMETER:  Origin-host-name              [264]
"diameter2.cisco.com"          (M)
*May  9 18:05:02.816: DIAMETER:  Origin-Realm                  [296]
"cisco.com"                    (M)
65913A20:                            01000050              ...P
65913A30: 0000011A 00000000 8AC75179 8AC75179  .........GQy.GQy
65913A40: 0000010C 4000000C 000007D1 00000108  ....@......Q....
65913A50: 4000001B 6469616D 65746572 322E6369  @...diameter2.ci
65913A60: 73636F2E 636F6D00 00000128 40000011  sco.com....(@...
65913A70: 63697363 6F2E636F 6D000000 00        cisco.com....
*May  9 18:05:02.816: Dia Base: Request message hash ctx removed for
[2328318329/2328318329]
*May  9 18:05:02.816: Dia Base: (C000000C): Received msg event from
message i/o
*May  9 18:05:02.816: Dia Peer FSM (50F63888): input event RCV_DPA in
state CLOSING
*May  9 18:05:02.816: Dia Base: (C000000C): Free the peer connection
context 50F63888
```

Related Commands

Command	Description
show diameter peer	Displays Diameter peer configuration information.

debug dlsw

To enable debugging of data-link switching plus (DLSw+), use the **debugdlsw**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dlsw border-peers [**interface interface**| **ip address ip-address**]**core** [**flow-control messages**| **state**| **xid**] **[circuit-number]local-circuit circuit-numberpeers** [**interface interface** [**fast-errors**| **fast-paks**]]**ip address ip-address** [**fast-errors**| **fast-paks**| **fst-seq**| **udp**]**reachability** [**error**| **verbose**] [**sna**| **netbios**]

no debug dlsw border-peers [**interface interface**| **ip address ip-address**]**core** [**flow-control messages**| **state**| **xid**] **[circuit-number]local-circuit circuit-numberpeers** [**interface interface** [**fast-errors**| **fast-paks**]]**ip address ip-address** [**fast-errors**| **fast-paks**| **fst-seq**| **udp**]**reachability** [**error**| **verbose**] [**sna**| **netbios**]

Syntax Description

border-peers	(Optional) Enables debugging output for border peer events.
interface *interface*	(Optional) Specifies a remote peer to debug by a direct interface.
ip address *ip-address*	(Optional) Specifies a remote peer to debug by its IP address.
core	(Optional) Enables debugging output for DLSw core events.
flow-control	(Optional) Enables debugging output for congestion in the WAN or at the remote end station.
messages	(Optional) Enables debugging output of core messages--specific packets received by DLSw either from one of its peers or from a local medium via the Cisco link services interface.
state	(Optional) Enables debugging output for state changes on the circuit.
xid	(Optional) Enables debugging output for the exchange identification state machine.
circuit-numbe r	(Optional) Specifies the circuit for which you want core debugging output to reduce the output.
local-circuit *circuit-number*	(Optional) Enables debugging output for circuits performing local conversion. Local conversion occurs when both the input and output data-link connections are on the same local peer and no remote peer exists.
peers	(Optional) Enables debugging output for peer events.

fast-errors	(Optional) Debugs errors for fast-switched packets.
fast-paks	(Optional) Debugs fast-switched packets.
fst-seq	(Optional) Debugs Fast-Sequenced Transport (FST) sequence numbers on fast switched packets.
udp	(Optional) Debugs User Datagram Protocol (UDP) packets.
reachability	(Optional) Enables debugging output for reachability events (explorer traffic). If no options are specified, event-level information is displayed for all protocols.
error |**verbose**	(Optional) Specifies how much reachability information you want displayed. The **verbose** keyword displays everything, including errors and events. The **error** keyword displays error information only. If no option is specified, event-level information is displayed.
sna | **netbios**	(Optional) Specifies that reachability information be displayed for only Systems Network Architecture (SNA) or Network Basic Input/Output System (NetBIOS) protocols. If no option is specified, information for all protocols is displayed.

Usage Guidelines

When you specify no optional keywords, the debug **dlsw** command enables all available DLSW debugging output.

Normally you need to use only the **error** or **verbose**option of the**debugdlswreachability**command to help identify problems. The **error** option is recommended for use by customers and provides a subset of the messages from the normal event-level debugging. The **verbose** option provides a very detailed view of events, and is typically used only by service personnel.

To reduce the amount of debug information displayed, use the **sna** or **netbios** option with the **debugdlswreachability**command if you know that you have an SNA or NetBIOS problem.

The DLSw core is the engine that is responsible for the establishment and maintenance of remote circuits. If possible, specifying the index of the specific circuit you want to debug reduces the amount of output displayed. However, if you want to watch a circuit initially come up, do not use the *circuit-number* option with the **core** keyword.

The **coreflow-control** option provides information about congestion in the WAN or at the remote end station. In these cases, DLSw sends Receiver Not Ready (RNR) frames on its local circuits, slowing data traffic on established sessions and giving the congestion an opportunity to clear.

The **corestate**option allows you to see when the circuit changes state. This capability is especially useful for determining why a session cannot be established or why a session is being disconnected.

The **coreXID**option allows you to track the exchange identification (XID)-state machine. The router tracks XID commands and responses used in negotiations between end stations before establishing a session.

Examples

The following examples show and explain some of the typical DLSw debugging messages you might see when using the**debugdlsw**command.

The following example enables UDP packet debugging for a specific remote peer:

```
Router# debug dlsw peers ip-address 1.1.1.6 udp
```

The following message is sample output from the **debugdlswborder-peers**command:

```
*Mar 10 17:39:56: CSM: delete group mac cache for group 0
*Mar 10 17:39:56: CSM: delete group name cache for group 0
*Mar 10 17:40:19: CSM: update group cache for mac 0000.3072.1070, group 10
*Mar 10 17:40:22: DLSw: send_to_group_members(): copy to peer 10.19.32.5
```

The following message is from a router that initiated a TCP connection:

```
DLSw: START-TPFSM (peer 10.3.8.7(2065)): event:ADMIN-OPEN CONNECTION state:DISCONN
DLSw: dtp_action_a() attempting to connect peer 10.3.8.7(2065)
DLSw: END-TPFSM (peer 10.3.8.7(2065)): state:DISCONN->WAIT_WR
DLSw: Async Open Callback 10.3.8.7(2065) -> 11002
DLSw: START-TPFSM (peer 10.3.8.7(2065)): event:TCP-WR PIPE OPENED state:WAIT_WR
DLSw: dtp_action_f() start read open timer for peer 10.3.8.7(2065)
DLSw: END-TPFSM (peer 10.3.8.7(2065)): state:WAIT_WR->WAIT_RD
DLSw: passive open 10.3.8.7(11004) -> 2065
DLSw: START-TPFSM (peer 10.3.8.7(2065)): event:TCP-RD PIPE OPENED state:WAIT_RD
DLSw: dtp_action_g() read pipe opened for peer 10.3.8.7(2065)
DLSw: CapExId Msg sent to peer 10.3.8.7(2065)
DLSw: END-TPFSM (peer 10.3.8.7(2065)): state:WAIT_RD->WAIT_CAP
DLSw: START-TPFSM (peer 10.3.8.7(2065)): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dtp_action_j() cap msg rcvd from peer 10.3.8.7(2065)
DLSw: Recv CapExId Msg from peer 10.3.8.7(2065)
DLSw: Pos CapExResp sent to peer 10.3.8.7(2065)
DLSw: END-TPFSM (peer 10.3.8.7(2065)): state:WAIT_CAP->WAIT_CAP
DLSw: START-TPFSM (peer 10.3.8.7(2065)): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dtp_action_j() cap msg rcvd from peer 10.3.8.7(2065)
DLSw: Recv CapExPosRsp Msg from peer 10.3.8.7(2065)
DLSw: END-TPFSM (peer 10.3.8.7(2065)): state:WAIT_CAP->WAIT_CAP
DLSw: Processing delayed event:SSP-CAP EXCHANGED - prev state:WAIT_CAP
DLSw: START-TPFSM (peer 10.3.8.7(2065)): event:SSP-CAP EXCHANGED state:WAIT_CAP
DLSw: dtp_action_k() cap xchged for peer 10.3.8.7(2065)
DLSw: closing read pipe tcp connection for peer 10.3.8.7(2065)
DLSw: END-TPFSM (peer 10.3.8.7(2065)): state:WAIT_CAP->PCONN_WT
DLSw: Processing delayed event:TCP-PEER CONNECTED - prev state:PCONN_WT
DLSw: START-TPFSM (peer 10.3.8.7(2065)): event:TCP-PEER CONNECTED state:PCONN_WT
DLSw: dtp_action_m() peer connected for peer 10.3.8.7(2065)
DLSw: END-TPFSM (peer 10.3.8.7(2065)): state:PCONN_WT->CONNECT
DLSw: START-TPFSM (peer 10.3.8.7(2065)): event:CORE-ADD CIRCUIT state:CONNECT
DLSw: dtp_action_u(), peer add circuit for peer 10.3.8.7(2065)
DLSw: END-TPFSM (peer 10.3.8.7(2065)): state:CONNECT->CONNECT
```

The following message is from a router that received a TCP connection:

```
DLSw: passive open 10.10.10.4(11002) -> 2065
DLSw: START-TPFSM (peer 10.10.10.4(2065)): event:TCP-RD PIPE OPENED state:DISCONN
DLSw: dtp_action_c() opening write pipe for peer 10.10.10.4(2065)
DLSw: END-TPFSM (peer 10.10.10.4(2065)): state:DISCONN->WWR_RDOP
DLSw: Async Open Callback 10.10.10.4(2065) -> 11004
DLSw: START-TPFSM (peer 10.10.10.4(2065)): event:TCP-WR PIPE OPENED state:WWR_RDOP
DLSw: dtp_action_i() write pipe opened for peer 10.10.10.4(2065)
DLSw: CapExId Msg sent to peer 10.10.10.4(2065)
DLSw: END-TPFSM (peer 10.10.10.4(2065)): state:WWR_RDOP->WAIT_CAP
DLSw: START-TPFSM (peer 10.10.10.4(2065)): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dtp_action_j() cap msg rcvd from peer 10.10.10.4(2065)
DLSw: Recv CapExId Msg from peer 10.10.10.4(2065)
DLSw: Pos CapExResp sent to peer 10.10.10.4(2065)
DLSw: END-TPFSM (peer 10.10.10.4(2065)): state:WAIT_CAP->WAIT_CAP
DLSw: START-TPFSM (peer 10.10.10.4(2065)): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dtp_action_j() cap msg rcvd from peer 10.10.10.4(2065)
DLSw: Recv CapExPosRsp Msg from peer 10.10.10.4(2065)
```

```
DLSw: END-TPFSM (peer 10.10.10.4(2065)): state:WAIT_CAP->WAIT_CAP
DLSw: Processing delayed event:SSP-CAP EXCHANGED - prev state:WAIT_CAP
DLSw: START-TPFSM (peer 10.10.10.4(2065)): event:SSP-CAP EXCHANGED state:WAIT_CAP
DLSw: dtp_action_k() cap xchged for peer 10.10.10.4(2065)
DLSw: END-TPFSM (peer 10.10.10.4(2065)): state:WAIT_CAP->PCONN_WT
DLSw: dlsw_tcpd_fini() for peer 10.10.10.4(2065)
DLSw: dlsw_tcpd_fini() closing write pipe for peer 10.10.10.4
DLSw: START-TPFSM (peer 10.10.10.4(2065)): event:TCP-CLOSE WR PIPE state:PCONN_WT
DLSw: dtp_action_l() close write pipe for peer 10.10.10.4(2065)
DLSw: closing write pipe tcp connection for peer 10.10.10.4(2065)
DLSw: END-TPFSM (peer 10.10.10.4(2065)): state:PCONN_WT->PCONN_WT
DLSw: Processing delayed event:TCP-PEER CONNECTED - prev state:PCONN_WT
DLSw: START-TPFSM (peer 10.10.10.4(2065)): event:TCP-PEER CONNECTED state:PCONN_WT
DLSw: dtp_action_m() peer connected for peer 10.10.10.4(2065)
DLSw: END-TPFSM (peer 10.10.10.4(2065)): state:PCONN_WT->CONNECT
DLSw: START-TPFSM (peer 10.10.10.4(2065)): event:CORE-ADD CIRCUIT state:CONNECT
DLSw: dtp_action_u(), peer add circuit for peer 10.10.10.4(2065)
DLSw: END-TPFSM (peer 10.10.10.4(2065)): state:CONNECT->CONNECT
```

The following message is from a router that initiated an FST connection:

```
DLSw: START-FSTPFSM (peer 10.10.10.4(0)): event:ADMIN-OPEN CONNECTION state:DISCONN
DLSw: dfstp_action_a() attempting to connect peer 10.10.10.4(0)
DLSw: Connection opened for peer 10.10.10.4(0)
DLSw: CapExId Msg sent to peer 10.10.10.4(0)
DLSw: END-FSTPFSM (peer 10.10.10.4(0)): state:DISCONN->WAIT_CAP
DLSw: START-FSTPFSM (peer 10.10.10.4(0)): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dfstp_action_e() cap msg rcvd for peer 10.10.10.4(0)
DLSw: Recv CapExPosRsp Msg from peer 10.10.10.4(0)
DLSw: END-FSTPFSM (peer 10.10.10.4(0)): state:WAIT_CAP->WAIT_CAP
DLSw: START-FSTPFSM (peer 10.10.10.4(0)): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dfstp_action_e() cap msg rcvd for peer 10.10.10.4(0)
DLSw: Recv CapExId Msg from peer 10.10.10.4(0)
DLSw: Pos CapExResp sent to peer 10.10.10.4(0)
DLSw: END-FSTPFSM (peer 10.10.10.4(0)): state:WAIT_CAP->WAIT_CAP
DLSw: Processing delayed event:SSP-CAP EXCHANGED - prev state:WAIT_CAP
DLSw: START-FSTPFSM (peer 10.10.10.4(0)): event:SSP-CAP EXCHANGED state:WAIT_CAP
DLSw: dfstp_action_f() cap xchged for peer 10.10.10.4(0)
DLSw: END-FSTPFSM (peer 10.10.10.4(0)): state:WAIT_CAP->CONNECT
```

The following message is from a router that received an FST connection:

```
DLSw: START-FSTPFSM (peer 10.3.8.7(0)): event:SSP-CAP MSG RCVD state:DISCONN
DLSw: dfstp_action_c() cap msg rcvd for peer 10.3.8.7(0)
DLSw: Recv CapExId Msg from peer 10.3.8.7(0)
DLSw: Pos CapExResp sent to peer 10.3.8.7(0)
DLSw: CapExId Msg sent to peer 10.3.8.7(0)
DLSw: END-FSTPFSM (peer 10.3.8.7(0)): state:DISCONN->WAIT_CAP
DLSw: START-FSTPFSM (peer 10.3.8.7(0)): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dfstp_action_e() cap msg rcvd for peer 10.3.8.7(0)
DLSw: Recv CapExPosRsp Msg from peer 10.3.8.7(0)
DLSw: END-FSTPFSM (peer 10.3.8.7(0)): state:WAIT_CAP->WAIT_CAP
DLSw: Processing delayed event:SSP-CAP EXCHANGED - prev state:WAIT_CAP
DLSw: START-FSTPFSM (peer 10.3.8.7(0)): event:SSP-CAP EXCHANGED state:WAIT_CAP
DLSw: dfstp_action_f() cap xchged for peer 10.3.8.7(0)
DLSw: END-FSTPFSM (peer 10.3.8.7(0)): state:WAIT_CAP->CONNECT
```

The following message is from a router that initiated an LLC2 connection:

```
DLSw-LLC2: Sending enable port ; port no : 0
           PEER-DISP Sent : CLSI Msg : ENABLE.Req   dlen: 20
DLSw: Peer Received : CLSI Msg : ENABLE.Cfm CLS_OK dlen: 20
DLSw-LLC2 : Sending activate sap for Serial1 - port_id = 887C3C
            port_type = 7 dgra(UsapID) = 952458
             PEER-DISP Sent : CLSI Msg : ACTIVATE_SAP.Req   dlen: 60
DLSw: Peer Received : CLSI Msg : ACTIVATE_SAP.Cfm CLS_OK dlen: 60
DLSw Got ActSapcnf back for Serial1 - port_id = 8978204, port_type = 7, psap_id = 0
DLSw: START-LLC2PFSM (peer on interface Serial1): event:ADMIN-OPEN CONNECTION state:DISCONN
DLSw: dllc2p_action_a() attempting to connect peer on interface Serial1
 PEER-DISP Sent : CLSI Msg : REQ_OPNSTN.Req   dlen: 106
DLSw: END-LLC2PFSM (peer on interface Serial1): state:DISCONN->ROS_SENT
DLSw: Peer Received : CLSI Msg : REQ_OPNSTN.Cfm CLS_OK dlen: 106
DLSw: START-LLC2PFSM (peer on interface Serial1): event:CLS-REQOPNSTN.CNF state:ROS_SENT
```

```
DLSw: dllc2p_action_c()
 PEER-DISP Sent : CLSI Msg : CONNECT.Req   dlen: 16
DLSw: END-LLC2PFSM (peer on interface Serial1): state:ROS_SENT->CON_PEND
DLSw: Peer Received : CLSI Msg : CONNECT.Cfm CLS_OK dlen: 28
DLSw: START-LLC2PFSM (peer on interface Serial1): event:CLS-CONNECT.CNF state:CON_PEND
DLSw: dllc2p_action_e() send capabilities to peer on interface Serial1
 PEER-DISP Sent : CLSI Msg : SIGNAL_STN.Req   dlen: 8
 PEER-DISP Sent : CLSI Msg : DATA.Req   dlen: 418
DLSw: CapExId Msg sent to peer on interface Serial1
DLSw: END-LLC2PFSM (peer on interface Serial1): state:CON_PEND->WAIT_CAP
DLSw: Peer Received : CLSI Msg : DATA.Ind   dlen: 418
DLSw: START-LLC2PFSM (peer on interface Serial1): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dllc2p_action_k() cap msg rcvd for peer on interface Serial1
DLSw: Recv CapExId Msg from peer on interface Serial1
 PEER-DISP Sent : CLSI Msg : DATA.Req   dlen: 96
DLSw: Pos CapExResp sent to peer on interface Serial1
DLSw: END-LLC2PFSM (peer on interface Serial1): state:WAIT_CAP->WAIT_CAP
DLSw: Peer Received : CLSI Msg : DATA.Ind   dlen: 96
DLSw: START-LLC2PFSM (peer on interface Serial1): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dllc2p_action_k() cap msg rcvd for peer on interface Serial1
DLSw: Recv CapExPosRsp Msg from peer on interface Serial1
DLSw: END-LLC2PFSM (peer on interface Serial1): state:WAIT_CAP->WAIT_CAP
DLSw: Processing delayed event:SSP-CAP EXCHANGED - prev state:WAIT_CAP
DLSw: START-LLC2PFSM (peer on interface Serial1): event:SSP-CAP EXCHANGED state:WAIT_CAP
DLSw: dllc2p_action_l() cap xchged for peer on interface Serial1
DLSw: END-LLC2PFSM (peer on interface Serial1): state:WAIT_CAP->CONNECT
```

The following message is from a router that received a Logical Link Control, type 2 (LLC2) connection:

```
DLSw-LLC2: Sending enable port ; port no : 0
 PEER-DISP Sent : CLSI Msg : ENABLE.Req   dlen: 20
DLSw: Peer Received : CLSI Msg : ENABLE.Cfm CLS_OK dlen: 20
DLSw-LLC2 : Sending activate sap for Serial0 - port_id = 887C3C
 port_type = 7 dgra(UsapID) = 93AB34
 PEER-DISP Sent : CLSI Msg : ACTIVATE_SAP.Req   dlen: 60
DLSw: Peer Received : CLSI Msg : ACTIVATE_SAP.Cfm CLS_OK dlen: 60
DLSw Got ActSapcnf back for Serial0 - port_id = 8944700, port_type = 7, psap_id = 0
DLSw: Peer Received : CLSI Msg : CONECT_STN.Ind   dlen: 39
DLSw: START-LLC2PFSM (peer on interface Serial0): event:CLS-CONNECT_STN.IND state:DISCONN
DLSw: dllc2p_action_s() conn_stn for peer on interface Serial0
 PEER-DISP Sent : CLSI Msg : REQ_OPNSTN.Req   dlen: 106
DLSw: END-LLC2PFSM (peer on interface Serial0): state:DISCONN->CONS_PEND
DLSw: Peer Received : CLSI Msg : REQ_OPNSTN.Cfm CLS_OK dlen: 106
DLSw: START-LLC2PFSM (peer on interface Serial0): event:CLS-REQOPNSTN.CNF state:CONS_PEND
DLSw: dllc2p_action_h() send capabilities to peer on interface Serial0
 PEER-DISP Sent : CLSI Msg : CONNECT.Rsp   dlen: 20
 PEER-DISP Sent : CLSI Msg : DATA.Req   dlen: 418
DLSw: CapExId Msg sent to peer on interface Serial0
DLSw: END-LLC2PFSM (peer on interface Serial0): state:CONS_PEND->WAIT_CAP
DLSw: Peer Received : CLSI Msg : CONNECTED.Ind   dlen: 8
DLSw: START-LLC2PFSM (peer on interface Serial0): event:CLS-CONNECTED.IND state:WAIT_CAP
DLSw: END-LLC2PFSM (peer on interface Serial0): state:WAIT_CAP->WAIT_CAP
DLSw: Peer Received : CLSI Msg : DATA.Ind   dlen: 418
DLSw: START-LLC2PFSM (peer on interface Serial0): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dllc2p_action_k() cap msg rcvd for peer on interface Serial0
DLSw: Recv CapExId Msg from peer on interface Serial0
 PEER-DISP Sent : CLSI Msg : DATA.Req   dlen: 96
DLSw: Pos CapExResp sent to peer on interface Serial0
DLSw: END-LLC2PFSM (peer on interface Serial0): state:WAIT_CAP->WAIT_CAP
DLSw: Peer Received : CLSI Msg : DATA.Ind   dlen: 96
DLSw: START-LLC2PFSM (peer on interface Serial0): event:SSP-CAP MSG RCVD state:WAIT_CAP
DLSw: dllc2p_action_k() cap msg rcvd for peer on interface Serial0
DLSw: Recv CapExPosRsp Msg from peer on interface Serial0
DLSw: END-LLC2PFSM (peer on interface Serial0): state:WAIT_CAP->WAIT_CAP
DLSw: Processing delayed event:SSP-CAP EXCHANGED - prev state:WAIT_CAP
DLSw: START-LLC2PFSM (peer on interface Serial0): event:SSP-CAP EXCHANGED state:WAIT_CAP
DLSw: dllc2p_action_l() cap xchged for peer on interface Serial0
DLSw: END-LLC2PFSM (peer on interface Serial0): state:WAIT_CAP->CONNECT
```

The following messages occur when a CUR_ex (CANUREACH explorer) frame is received from other peers, and the peer statements or the **promiscuous** keyword have not been enabled so that the router is not configured correctly:

```
22:42:44: DLSw: Not promiscuous - Rej conn from 172.20.96.1(2065)
22:42:51: DLSw: Not promiscuous - Rej conn from 172.20.99.1(2065)
```

In the following messages, the router sends a keepalive message every 30 seconds to keep the peer connected. If three keepalive messages are missed, the peer is torn down. These messages are displayed only if keepalives are enabled (by default, keepalives are disabled):

```
22:44:03: DLSw: Keepalive Request sent to peer 172.20.98.1(2065) (168243148)
22:44:03: DLSw: Keepalive Response from peer 172.20.98.1(2065) (168243176)
22:44:34: DLSw: Keepalive Request sent to peer 172.20.98.1(2065) (168274148)
22:44:34: DLSw: Keepalive Response from peer 172.20.98.1(2065) (168274172)
```

The following peer debugging messages indicate that the local peer is disconnecting from the specified remote peer because of missed peer keepalives:

```
0:03:24: DLSw: keepalive failure for peer on interface Serial0
0:03:24: DLSw: action_d(): for peer on interface Serial0
0:03:24: DLSW: DIRECT aborting connection for peer on interface Serial0
0:03:24: DLSw: peer on interface Serial0, old state CONNECT, new state DISCONN
```

The following peer debugging messages result from an attempt to connect to an IP address that does not have DLSw enabled. The local router attempts to connect in 30-second intervals:

```
23:13:22: action_a() attempting to connect peer 172.20.100.1(2065)
23:13:22: DLSw: CONN: peer 172.20.100.1 open failed, rejected [9]
23:13:22: action_a() retries: 8 next conn time: 861232504
23:13:52: action_a() attempting to connect peer 172.20.100.1(2065)
23:13:52: DLSw: CONN: peer 172.20.100.1 open failed, rejected [9]
23:13:52: action_a() retries: 9 next conn time: 861292536
```

The following peer debugging messages that indicates a remote peer statement is missing on the router (address 172.20.100.1) to which the connection attempt is sent:

```
23:14:52: action_a() attempting to connect peer 172.20.100.1(2065)
23:14:52: DLSw: action_a(): Write pipe opened for peer 172.20.100.1(2065)
23:14:52: DLSw: peer 172.20.100.1(2065), old state DISCONN, new state WAIT_RD
23:14:52: DLSw: dlsw_tcpd_fini() closing connection for peer 172.20.100.1
23:14:52: DLSw: action_d(): for peer 172.20.100.1(2065)
23:14:52: DLSw: aborting tcp connection for peer 172.20.100.1(2065)
23:14:52: DLSw: peer 172.20.100.1(2065), old state WAIT_RD, new state DISCONN
```

The following messages show a peer connection opening with no errors or abnormal events:

```
23:16:37: action_a() attempting to connect peer 172.20.100.1(2065)
23:16:37: DLSw: action_a(): Write pipe opened for peer 172.20.100.1(2065)
23:16:37: DLSw: peer 172.20.100.1(2065), old state DISCONN, new state WAIT_RD
23:16:37: DLSW: passive open 172.20.100.1(17762) -> 2065
23:16:37: DLSw: action_c(): for peer 172.20.100.1(2065)
23:16:37: DLSw: peer 172.20.100.1(2065), old state WAIT_RD, new state CAP_EXG
23:16:37: DLSw: peer 172.20.100.1(2065) conn_start_time set to 861397784
23:16:37: DLSw: CapExId Msg sent to peer 172.20.100.1(2065)
23:16:37: DLSw: Recv CapExId Msg from peer 172.20.100.1(2065)
23:16:37: DLSw: Pos CapExResp sent to peer 172.20.100.1(2065)
23:16:37: DLSw: action_e(): for peer 172.20.100.1(2065)
23:16:37: DLSw: Recv CapExPosRsp Msg from peer 172.20.100.1(2065)
23:16:37: DLSw: action_e(): for peer 172.20.100.1(2065)
23:16:37: DLSw: peer 172.20.100.1(2065), old state CAP_EXG, new state CONNECT
23:16:37: DLSw: dlsw_tcpd_fini() closing write pipe for peer 172.20.100.1
23:16:37: DLSw: action_g(): for peer 172.20.100.1(2065)
23:16:37: DLSw: closing write pipe tcp connection for peer 172.20.100.1(2065)
23:16:38: DLSw: peer_act_on_capabilities() for peer 172.20.100.1(2065)
```

The following two messages show that an information frame is passing through the router:

```
DLSw: dlsw_tr2fct() lmac:c000.a400.0000 rmac:0800.5a29.75fe ls:5 rs:4 i:34
DLSw: dlsw_tr2fct() lmac:c000.a400.0000 rmac:0800.5a29.75fe ls:4 rs:4 i:34
```

Examples

The messages in this section are based on the following criteria:

- Reachability is stored in cache. DLSw+ maintains two reachability caches: one for MAC addresses and one for NetBIOS names. Depending on how long entries have been in the cache, they are either fresh or stale.
- If a router has a fresh entry in the cache for a certain resource, it answers a locate request for that resource without verifying that it is still available. A locate request is typically a TEST frame for MAC addresses or a FIND_NAME_QUERY for NetBIOS.
- If a router has a stale entry in the cache for a certain resource, it verifies that the entry is still valid before answering a locate request for the resource by sending a frame to the last known location of the resource and waits for a resource. If the entry is a REMOTE entry, the router sends a CUR_ex frame to the remote peer to verify. If the entry is a LOCAL entry, it sends either a TEST frame or a NetBIOS FIND_NAME_QUERY on the appropriate local port.
- By default, all reachability cache entries remain fresh for 4 minutes after they are learned. For MAC addresses, you can change this time with the **dlswtimersna-verify-interval** command. For NetBIOS names, you can change this time with the **dlswtimernetbios-verify-interval** command.
- By default, all reachability cache entries age out of the cache 16 minutes after they are learned. For MAC addresses, you can change this time with the **dlswtimersna-cache-timeout** command. For NetBIOS names, you can change the time with the **dlswtimernetbios-cache-timeout** command.

The table below describes the debug output indicating that the DLSW router received an SSP message that is flow controlled and should be counted against the window of the sender.

```
Dec  6 11:26:49: CSM: Received SSP  CUR   csex flags = 80, mac 4000.90b1.26cf,
The csex flags = 80 means that this is an CUR_ex (explorer).
Dec  5 10:48:33: DLSw: 1620175180 decr r - s:27 so:0 r:27 ro:0
```

Table 6: debug dlsw Field Descriptions

Field	Description
decr r	Decrement received count.
s	This DLSW router's granted units for the circuit.
so	0=This DLSW router does not owe a flow control acknowledgment. 1=This router owes a flow control acknowledgment.
r	Partner's number of granted units for the circuit.
ro	Indicates whether the partner owes flow control acknowledgment.

The following message shows that DLSW is sending an I frame to a LAN:

```
Dec  5 10:48:33:  DISP Sent : CLSI Msg : DATA.Req   dlen: 1086
```

The following message shows that DLSW received the I frame from the LAN:

```
Dec  5 10:48:35:  DLSW Received-disp : CLSI Msg : DATA.Ind   dlen: 4
```

The following messages show that the reachability cache is cleared:

```
Router# clear dlsw rea
23:44:11: CSM: Clearing CSM cache
23:44:11: CSM: delete local mac cache for port 0
23:44:11: CSM: delete local name cache for port 0
23:44:11: CSM: delete remote mac cache for peer 0
23:44:11: CSM: delete remote name cash dlsw rea
```

The next group of messages show that the DLSW reachability cache is added, and that a name query is perform from the router MARIAN:

```
23:45:11: CSM: core_to_csm CLSI_MSG_PROC - port_id 5EFBB4
23:45:11: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:11: CSM: update local cache for mac 0800.5a30.7a9b, port 5EFBB4
23:45:11: CSM: update local cache for name MARIAN         , port 5EFBB4
23:45:11: CSM: Received CLS_UDATA_STN from Core
23:45:11: CSM: Received netbios frame type A
23:45:11: CSM: Processing Name Query
23:45:11: CSM: Netbios Name Query: ws_status = 6
23:45:11: CSM: Write to peer 0 ok.
23:45:11: CSM: Freeing clsi message
23:45:11: CSM: core_to_csm CLSI_MSG_PROC - port_id 658AB4
23:45:11: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:11: CSM: update local cache for mac 0800.5a30.7a9b, port 658AB4
23:45:11: CSM: update local cache for name MARIAN         , port 658AB4
23:45:11: CSM: Received CLS_UDATA_STN from Core
23:45:11: CSM: Received netbios frame type A
23:45:11: CSM: Processing Name Query
23:45:11: CSM: Netbios Name Query: ws_status = 5
23:45:11: CSM: DLXNR_PEND match found.... drop name query
23:45:11: CSM: Freeing clsi message
23:45:12: CSM: core_to_csm CLSI_MSG_PROC - port_id 5EFBB4
23:45:12: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:12: CSM: update local cache for mac 0800.5a30.7a9b, port 5EFBB4
23:45:12: CSM: update local cache for name MARIAN         , port 5EFBB4
23:45:12: CSM: Received CLS_UDATA_STN from Core
23:45:12: CSM: Received netbios frame type A
23:45:12: CSM: Processing Name Query
23:45:12: CSM: Netbios Name Query: ws_status = 5
23:45:12: CSM: DLXNR_PEND match found.... drop name query
23:45:12: CSM: Freeing clsi message
23:45:12: CSM: core_to_csm CLSI_MSG_PROC - port_id 658AB4
23:45:12: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:12: CSM: update local cache for mac 0800.5a30.7a9b, port 658AB4
23:45:12: CSM: update local cache for name MARIAN         , port 658AB4
23:45:12: CSM: Received CLS_UDATA_STN from Core
23:45:12: CSM: Received netbios frame type A
23:45:12: CSM: Processing Name Query
23:45:12: CSM: Netbios Name Query: ws_status = 5
23:45:12: CSM: DLXNR_PEND match found.... drop name query
23:45:12: CSM: Freeing clsi message
23:45:12: CSM: core_to_csm CLSI_MSG_PROC - port_id 5EFBB4
23:45:12: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:12: CSM: update local cache for mac 0800.5a30.7a9b, port 5EFBB4
23:45:12: CSM: update local cache for name MARIAN         , port 5EFBB4
23:45:12: CSM: Received CLS_UDATA_STN from Core
23:45:12: CSM: Received netbios frame type A
23:45:12: CSM: Processing Name Query
23:45:12: CSM: Netbios Name Query: ws_status = 5
23:45:12: CSM: DLXNR_PEND match found.... drop name query
23:45:12: CSM: Freeing clsi message
23:45:12: CSM: core_to_csm CLSI_MSG_PROC - port_id 658AB4
```

```
23:45:12: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:12: CSM: update local cache for mac 0800.5a30.7a9b, port 658AB4
23:45:12: CSM: update local cache for name MARIAN         , port 658AB4
23:45:12: CSM: Received CLS_UDATA_STN from Core
23:45:12: CSM: Received netbios frame type A
23:45:12: CSM: Processing Name Query
23:45:12: CSM: Netbios Name Query: ws_status = 5
23:45:12: CSM: DLXNR_PEND match found.... drop name query
23:45:12: CSM: Freeing clsi message
23:45:18: CSM: Deleting Reachability cache
23:45:18: CSM: Deleting DLX NR pending record....
23:45:38: CSM: core_to_csm CLSI_MSG_PROC - port_id 5EFBB4
23:45:38: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:38: CSM: update local cache for mac 0800.5a30.7a9b, port 5EFBB4
23:45:38: CSM: update local cache for name MARIAN         , port 5EFBB4
23:45:38: CSM: Received CLS_UDATA_STN from Core
23:45:38: CSM: Received netbios frame type 8
23:45:38: CSM: Write to peer 0 ok.
23:45:38: CSM: Freeing clsi message
23:45:38: CSM: core_to_csm CLSI_MSG_PROC - port_id 658AB4
23:45:38: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:38: CSM: update local cache for mac 0800.5a30.7a9b, port 658AB4
23:45:38: CSM: update local cache for name MARIAN         , port 658AB4
23:45:38: CSM: Received CLS_UDATA_STN from Core
23:45:38: CSM: Received netbios frame type 8
23:45:38: CSM: Write to peer 0 ok.
23:45:38: CSM: Freeing clsi message
```

The following messages show that the router named MARIAN is added to the network:

```
23:45:38: CSM: core_to_csm CLSI_MSG_PROC - port_id 5EFBB4
23:45:38: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:38: CSM: update local cache for mac 0800.5a30.7a9b, port 5EFBB4
23:45:38: CSM: update local cache for name MARIAN         , port 5EFBB4
23:45:38: CSM: Received CLS_UDATA_STN from Core
23:45:38: CSM: Received netbios frame type 8
23:45:38: CSM: Write to peer 0 ok.
23:45:38: CSM: Freeing clsi message
23:45:38: CSM: core_to_csm CLSI_MSG_PROC - port_id 658AB4
23:45:38: CSM: 0800.5a30.7a9b passes local mac excl. filter
23:45:38: CSM: update local cache for mac 0800.5a30.7a9b, port 658AB4
23:45:38: CSM: update local cache for name MARIAN         , port 658AB4
23:45:38: CSM: Received CLS_UDATA_STN from Core
23:45:38: CSM: Received netbios frame type 8
23:45:38: CSM: Write to peer 0 ok.
23:45:38: CSM: Freeing clsi message
```

In the next group of messages, an attempt is made to add the router named GINGER on the Ethernet interface:

```
0:07:44: CSM: core_to_csm CLSI_MSG_PROC - port_id 658AB4
0:07:44: CSM: 0004.f545.24e6 passes local mac excl. filter
0:07:44: CSM: update local cache for mac 0004.f545.24e6, port 658AB4
0:07:44: CSM: update local cache for name GINGER         , port 658AB4
0:07:44: CSM: Received CLS_UDATA_STN from Core
0:07:44: CSM: Received netbios frame type 8
0:07:44: CSM: Write to peer 0 ok.
```

In the following example, the output from the **showdlswreachability**command indicates that GINGER is on the Ethernet interface and MARIAN is on the Token Ring interface:

```
Router# show dlsw reachability
DLSw MAC address reachability cache list
Mac Addr        status     Loc.    peer/port            rif
0004.f545.24e6  FOUND      LOCAL   P007-S000    --no rif--
0800.5a30.7a9b  FOUND      LOCAL   P000-S000    06C0.0621.7D00
                                   P007-S000    F0F8.0006.A6FC.005F.F100.0000.0000.0000
DLSw NetBIOS Name reachability cache list
NetBIOS Name    status     Loc.    peer/port            rif
GINGER          FOUND      LOCAL   P007-S000     --no rif--
MARIAN          FOUND      LOCAL   P000-S000     06C0.0621.7D00
                                   P007-S000     --no rif--
```

debug dmsp doc-to-fax

Note In release 12.3(8)T, the **debugdmspdoc-to-fax** command is replaced by the **debugfaxdmsp** command. See the **debugfaxdmsp** command for more information.

To display debugging messages for the doc Media Service Provider (docMSP) TIFF or text2Fax engine, use the **debugdmspdoc-to-fax**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dmsp doc-to-fax [**text-to-fax**| **tiff-reader**]

no debug dmsp doc-to-fax [**text-to-fax**| **tiff-reader**]

Syntax Description

text-to-fax	(Optional) Displays debugging messages that occur while the DocMSP Component is receiving text packets and producing T4 fax data.
tiff-reader	(Optional) Displays debugging messages that occur while the DocMSP Component is receiving TIFF packets and producing T4 fax data.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XI	This command was introduced on the Cisco AS5300 access server.
12.3(8)T	This command was replaced by the **debugfaxdmsp** command in the Cisco IOS 12.3T release.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debugdmspdoc-to-fax** command:

```
Router# debug dmsp doc-to-fax
Jan  1 04:58:39.898: docmsp_call_setup_request: callid=18
Jan  1 04:58:39.902: docmsp_call_setup_request():  ramp data dir=OFFRAMP, conf dir=SRC
Jan  1 04:58:39.902: docmsp_caps_ind: call id=18, src=17
Jan  1 04:58:39.902: docmsp_bridge cfid=5, srccid=18, dstcid=17
Jan  1 04:58:39.902: docmsp_bridge(): ramp data dir=OFFRAMP, conf dir=SRC, encode out=2
Jan  1 04:58:39.902: docmsp_rcv_msp_ev: call id =18, evID = 42
Jan  1 04:58:39.902: docmsp_bridge cfid=6, srccid=18, dstcid=15
Jan  1 04:58:39.902: docmsp_bridge(): ramp data dir=OFFRAMP, conf dir=DEST, encode out=2
Jan  1 04:58:39.902: docmsp_process_rcv_data: call id src=0, dst=18
Jan  1 04:58:39.902: docmsp_generate_page:
Jan  1 04:58:39.902: docmsp_generate_page: new context for Call 18
Jan  1 04:58:39.922: docmsp_get_msp_event_buffer:
Jan  1 04:58:42.082: docmsp_xmit: call id src=15, dst=18
Jan  1 04:58:42.082: docmsp_process_rcv_data: call id src=15, dst=18
Jan  1 04:58:42.082: offramp_data_process:
Jan  1 04:58:42.102: docmsp_xmit: call id src=15, dst=18
Jan  1 04:58:42.106: docmsp_process_rcv_data: call id src=15, dst=18
Jan  1 04:58:42.106: offramp_data_process:
Jan  1 04:58:42.122: docmsp_xmit: call id src=15, dst=18
Jan  1 04:58:42.126: docmsp_process_rcv_data: call id src=15, dst=18
Jan  1 04:58:42.126: offramp_data_process:
Jan  1 04:58:42.142: docmsp_xmit: call id src=15, dst=18
Jan  1 04:58:42.146: docmsp_xmit: call id src=15, dst=18
```

Related Commands

Command	Description
debug dmsp fax-to-doc	Displays debugging messages for doc MPS fax-to-doc.

debug dmsp fax-to-doc

Note In release 12.3(8)T, the **debugdmspfax-to-doc**command is replaced by the **debugfaxdmsp** command. See the **debugfaxdmsp** command for more information.

To display debugging messages for doc MSP (docMSP) fax-to-doc, use the **debugdmspfax-to-doccommandinprivileged**EXEC mode. To disable debugging output, use the **no** form of this command.

debug dmsp fax-to-doc [tiff-writer]

no debug dmsp fax-to-doc [tiff-writer]

Syntax Description

tiff-writer	(Optional) Displays debug messages that occur while the DocMSP Component is receiving T4 fax data and producing TIFF packets.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XI	This command was introduced on the Cisco AS5300 access server.
12.3(8)T	This command was replaced by the **debugfaxdmsp** command in the Cisco IOS 12.3T release.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debugdmspfax-to-doc** command:

```
Router# debug dmsp fax-to-doc
*Oct 16 08:29:54.487: docmsp_call_setup_request: callid=22
*Oct 16 08:29:54.487: docmsp_call_setup_request():  ramp data dir=OFFRAMP, conf dir=SRC
*Oct 16 08:29:54.487: docmsp_caps_ind: call id=22, src=21
*Oct 16 08:29:54.487: docmsp_bridge cfid=15, srccid=22, dstcid=21
*Oct 16 08:29:54.487: docmsp_bridge(): ramp data dir=OFFRAMP, conf dir=SRC, encode out=2
*Oct 16 08:29:54.487: docmsp_bridge cfid=16, srccid=22, dstcid=17
*Oct 16 08:29:54.487: docmsp_bridge(): ramp data dir=OFFRAMP, conf dir=DEST, encode out=2
*Oct 16 08:29:54.487: docmsp_xmit: call id src=17, dst=22
*Oct 16 08:29:54.487: docmsp_process_rcv_data: call id src=17, dst=22
*Oct 16 08:29:54.487: offramp_data_process:
```

```
*Oct 16 08:29:54.515: docmsp_get_msp_event_buffer:
*Oct 16 08:29:56.115: docmsp_call_setup_request: callid=24
*Oct 16 08:29:56.115: docmsp_call_setup_request():  ramp data dir=ONRAMP, conf dir=DEST
*Oct 16 08:29:56.115: docmsp_caps_ind: call id=24, src=20
*Oct 16 08:29:56.115: docmsp_bridge cfid=17, srccid=24, dstcid=20
```

Related Commands

Command	Description
debug dmsp doc-to-fax	Displays debugging messages for the doc Media Service Provider TIFF or text2Fax engine.

debug dmvpn

To display debug Dynamic Multipoint VPN (DMVPN) session information, use the **debug dmvpn** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dmvpn {**all**| **error**| **detail**| **packet**} {**all**| *debug-type*}

no debug dmvpn {**all**| **error**| **detail**| **packet**} {**all**| *debug-type*}

Syntax Description

all	Enables all levels of debugging.
error	Enables error-level debugging.
detail	Enables detail-level debugging.
packet	Enables packet-level debugging.
all	Enables NHRP, sockets, tunnel protection, and crypto debugging.
debug-type	The type of debugging that you want to enable. The following keywords can be specified for the *debug-type* argument: • **nhrp** -- Enables Next Hop Resolution Protocol (NHRP) debugging only. • **crypto** -- Enables crypto Internet Key Exchange (IKE) and IPsec debugging. • **tunnel** -- Enables tunnel protection debugging. • **socket** -- Enables crypto secure socket debugging. The keywords can be used alone, or in any combination with each other, but each keyword can be used only once.

Command Default

DMVPN debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.

Release	Modification
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
Cisco IOS XE Release 2.5	This command was modified. This command was integrated into Cisco IOS XE Release 2.5.

Usage Guidelines

You must specify both the level and the type of debugging that you want to enable. The debugging levels are all, error, detail, or packet. You can enable NHRP, crypto Internet Key Exchange (IKE) and IPsec, tunnel protection, and crypto secure socket debugging at any of the four debugging levels.

To enable conditional DMVPN debugging, you must first specify the level and type of debugging that you want to enable, and then use the **debug dmvpn condition** command to specify the conditions that you want to enable.

Error-Level Debugging

When error-level debugging is enabled with the **debug dmvpn error**command, the following debugging commands are enabled by default:

- **debug crypto ipsec error**
- **debug crypto isakmp error**
- **debug nhrp error**

Detail-Level Debugging

When detail-level debugging is enabled with the **debug dmvpn detail**command, the following debugging commands are enabled by default:

- **debug crypto ipsec**
- **debug crypto isakmp**
- **debug crypto sockets**
- **debug nhrp**
- **debug nhrp cache**
- **debug nhrp rate**
- **debug tunnel protection**

Packet-Level Debugging

When packet-level debugging is enabled with the **debug dmvpn packet**command, the following debugging commands are enabled by default:

- **debug nhrp extension**
- **debug nhrp packet**

Note Executing the **debug dmvpn all** command with a high number of active sessions may result in high CPU utilization and large data output.

NHRP Shortcut Route Debugging

When shortcut switching is enabled on the router, the system looks up the NHRP shortcut route in the Routing Information Base (RIB) in order to forward the packet to the next-hop in the DMVPN cloud.

The table below describes the debug messages displayed by the router when shortcut switching and NHRP debugging are both enabled.

Table 7: Sample Messages for Shortcut Switching and NHRP

Event	Sample Message

NHRP successfully adds a route to the RIB	`*Feb 21 13:11:24.043: NHRP: Adding route entry for 172.16.99.0 to RIB`
	`*Feb 21 13:11:24.043: NHRP: Route addition to RIB successful`

NHRP is unable to add a route to the RIB	`*Feb 21 13:11:24.043: NHRP: Adding route entry for 172.16.99.0 to RIB`
	`*Feb 21 13:11:24.043: NHRP: Route addition to RIB failed`

NHRP removes a route from the RIB	`*Feb 21 13:11:24.043: NHRP: Deleting route entry for 172.16.99.0 from RIB`
NHRP evicts a route from the RIB	`*Mar 1 18:24:29.371: NHRP: Route entry 172.16.22.0/24 clobbered by distance`
NHRP changes the administrative distance	`*Mar 1 00:14:16.799: NHRP: Administrative distance changed to 240`

Examples

The following example shows how to enable all debugging levels for DMVPN tunnel debugging:

```
Router# debug dmvpn all tunnel
```

Related Commands

Command	Description
debug crypto error	Enables error debugging for a crypto area.
debug crypto ipsec	Displays IPsec events.
debug crypto isakmp	Displays messages about IKE events.
debug dmvpn condition	Display conditional debug DMVPN session information.

Command	Description
debug nhrp condition	Enables NHRP conditional debugging.
debug nhrp error	Displays NHRP error-level debugging information.

debug dmvpn condition

To display conditional debug Dynamic Multipoint VPN (DMVPN) session information, use the **debug dmvpn condition**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dmvpn condition {**unmatched**| **peer** {**nbma**| **tunnel** {*ipv4-address*| *ipv6-address*}}| **vrf** *vrf-name*| **interface tunnel** *tunnel-interface*}

no debug dmvpn condition [**unmatched**| **peer** {**nbma**| **tunnel** {*ipv4-address*| *ipv6-address*}}| **vrf** *vrf-name*| **interface tunnel** *number*]

Syntax Description

unmatched	Specifies debugging when context information is not available.
peer	Specifies information for a specific DMVPN peer.
nbma	Displays DMVPN information based on the peer mapping nonbroadcast access (NBMA) address.
tunnel	Displays DMVPN information based on the peer Virtual Private Network (VPN) address.
ipv4-address	The DMVPN peer IPv4 address.
ipv6-address	The DMVPN peer IPv6 address. **Note** Cisco IOS XE Release 2.5 does not support the ipv6-address argument.
vrf	Displays information based on the specified virtual routing and forwarding (VRF) name.
vrf-name	The VRF name.
interface	Displays DMVPN information based on a specific interface.
tunnel	Specifies the tunnel address for a DMVPN peer.
number	The tunnel interface number.

Command Default

DMVPN conditional debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	The *ipv6-address* argument was added.
Cisco IOS XE Release 2.5	This command was modified. It was integrated into Cisco IOS XE Release 2.5.

Usage Guidelines

Conditional debugging is enabled only after the DMVPN debugging type and level have been specified using the **debug dmvpn** command.

Console Output

The following **debug dmvpn** commands do not have any console output on the Cisco 3845 and Cisco 7200 series routers:

- **debug dmvpn condition interface**
- **debug dmvpn condition peer**
- **debug dmvpn condition unmatched**
- **debug dmvpn condition vrf**

Note When the **debug dmvpn condition unmatched** command is enabled on the Cisco 3845 and Cisco 7200 series routers, issuing the **show debugging** command does not produce any console output.

Examples

The following example shows how to enable conditional DMVPN debugging for a specific peer NBMA address:

```
Router# debug dmvpn condition peer nbma 192.0.2.1
```

The following example shows how to enable conditional DMVPN debugging when context is not available to check against debugging conditions:

```
Router# debug dmvpn condition unmatched
```

The following example shows how to disable conditional debugging for a specific tunnel interface:

```
Router# no debug dmvpn condition interface tunnel 1
```

The following example shows how to disable all conditional debugging:

```
Router# no debug dmvpn condition
```

Related Commands

Command	Description
debug crypto error	Enables error debugging for a crypto area.
debug crypto ipsec	Displays IPsec events.
debug crypto isakmp	Displays messages about IKE events.
debug dmvpn	Displays debug DMVPN session information.
debug nhrp condition	Enables NHRP conditional debugging.
debug nhrp error	Displays NHRP error-level debugging information.

debug dot11

To enable debugging of radio functions, use the **debugdot11**command in privileged EXEC mode. To stop or disable the debug operation, use the **no** form of this command.

debug dot11 {**events**| **forwarding**| **mgmt**| **packets**| **syslog**| **virtual-interface**}

no debug dot11 {**events**| **forwarding**| **mgmt**| **packets**| **syslog**| **virtual-interface**}

Syntax Description

events	Displays information about all radio-related events.
forwarding	Displays information about radio-forwarded packets.
mgmt	Displays information about radio access point management activity.
packets	Displays information about received or transmitted radio packets.
syslog	Displays information about the radio system log.
virtual-interface	Displays information about radio virtual interfaces.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(4)JA	This command was introduced.
12.4(2)T	This command was integrated into Cisco IOS Release 12.4(2)T.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to display debugging information about radio functions.

Examples

The following example shows how to enable debugging of all radio-related events:

```
Router# debug dot11 events
```

Related Commands

Command	Description
debug dot11 aaa	Enables debugging of dot11 AAA operations.
debug dot11 dot11radio	Enables radio debug options.

debug dot11 aaa

To enable debugging of dot11 authentication, authorization, and accounting (AAA) operations, use the **debugdot11aaa**command in privileged EXEC mode. To disable or stop the debug operation, use the **no** form of this command.

debug dot11 aaa {**accounting**| **authenticator** {**all**| **dispatcher**| **mac-authen**| **process**| **rxdata**| **state-machine**| **txdata**}| **dispatcher**| **manager** {**all**| **dispatcher**| **keys**| **rxdata**| **state-machine**| **supplicant**| **txdata**}}

no debug dot11 aaa {**accounting**| **authenticator** {**all**| **dispatcher**| **mac-authen**| **process**| **rxdata**| **state-machine**| **txdata**}| **dispatcher**| **manager** {**all**| **dispatcher**| **keys**| **rxdata**| **state-machine**| **supplicant**| **txdata**}}

Syntax Description

accounting	Provides information about 802.11 AAA accounting packets.
authenticator	Provides information about MAC and Extensible Authentication Protocol (EAP) authentication packets. Use the following options to activate authenticator debugging: • all--Activates debugging for all authenticator packets • dispatcher--Activates debugging for authentication request handler packets • mac-authen--Activates debugging for MAC authentication packets • process--Activates debugging for authenticator process packets • rxdata--Activates debugging for EAP over LAN (EAPOL) packets from client devices • state-machine--Activates debugging for authenticator state-machine packets • txdata--Activates debugging for EAPOL packets sent to client devices
dispatcher	Provides information about 802.11 AAA dispatcher (interface between association and manager) packets.

manager	Provides information about the AAA manager. Use these options to activate AAA manager debugging: • **all** --Activates all AAA manager debugging • **dispatcher** --Activates debug information for AAA manager-authenticator dispatch traffic • **keys** --Activates debug information for AAA manager key processing • **rxdata** --Activates debugging for AAA manager packets received from client devices • **state-machine** --Activates debugging for AAA manager state-machine packets • **supplicant** --Activates debugging for Light Extensible Authentication Protocol (LEAP) supplicant packets • **txdata** --Activates debugging for AAA manager packets sent to client devices.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(4)JA	This command was introduced.
12.2(15)JA	This command was modified to include the accounting, authenticator, dispatcher, and manager debugging options.
12.4(2)T	This command was integrated into Cisco IOS Release 12.4(2)T.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to display debugging information about dot11 AAA operations.

Examples

The following example shows how to activate debugging for 802.11 AAA accounting packets:

```
Router# debug dot11 aaa accounting
```

Related Commands

Command	Description
debug dot11	Enables debugging of radio functions.
debug dot11 dot11radio	Enables radio debug options.

debug dot11 cac

Use the **debugdot11cac** privileged EXEC command to begin debugging of admission control radio functions. Use the **no**form of this command to stop the debug operation.

[no] debug dot11 cac {**events**| **unit**}

Syntax Description

events	Activates debugging of radio admission control events.
unit	Activates verbose debugging of radio admission control events.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(8)JA	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

This example shows how to begin debugging of all admission control radio-related events:

```
SOAP-AP# debug dot11 cac events
```

This example shows how to begin verbose debugging of all admission control radio-related events:

```
SOAP-AP# debug dot11 cac unit
```

This example shows how to stop debugging of all admission control radio-related events:

```
SOAP-AP# debug dot11 cac events
```

This example shows how to stop verbose debugging of all admission control radio-related events:

```
SOAP-AP# no debug dot11 cac unit
```

Related Comma

Note This command is not supported on repeaters.

Command	Description
admin-traffic (SSID configuration mode)	Enables CAC admission control for an SSID on the access point.
admit-traffic (QOS Class interface configuration mode)	Configures CAC admission control on the access point.
show debugging	Displays all debug settings and the debug packet headers
show dot11 ids eap	Displays all CAC radio events on the access point.
traffic-stream	Configures CAC traffic data rates and priorities for a radio interface on the access point.

debug dot11 dot11radio

To enable radio debug options, use the **debugdot11dot11radio**command in privileged EXEC mode. To disable debug options, use the **no** form of this command.

debug dot11 dot11radio *interface* {**accept-radio-firmware**| **dfs simulate** [*channel*]| **monitor** {**ack**| **address**| **beacon**| **crc**| **lines**| **plcp**| **print**| **probe**| **store**}| **print** {**hex**| **if**| **iv**| **lines**| **mic**| **plcp**| **printf**| **raw**| **shortadr**}| **stop-on-failure**| **trace** {**off**| **print**| **store**}}

no debug dot11 dot11radio *interface* {**accept-radio-firmware**| **dfs simulate** [*channel*]| **monitor** {**ack**| **address**| **beacon**| **crc**| **lines**| **plcp**| **print**| **probe**| **store**}| **print** {**hex**| **if**| **iv**| **lines**| **mic**| **plcp**| **printf**| **raw**| **shortadr**}| **stop-on-failure**| **trace** {**off**| **print**| **store**}}

Syntax Description

interface	The radio interface. The 2.4-GHz radio is 0. The 5-GHz radio is 1.
accept-radio-firmware	Configures the access point to disable checking the radio firmware version.
dfs simulate	Configures the access point to simulate radar generation as part of Dynamic Frequency Selection (DFS).
channel	(Optional) Radio channel to move to. Range is from 24 to 161.
monitor	Enables RF monitor mode. Use these options to turn on monitor modes: • **ack** --Displays ACK packets. ACK packets acknowledge receipt of a signal, information, or packet. • **address** --Displays packets to or from the specified IP address • **beacon** --Displays beacon packets • **crc** --Displays packets with CRC errors • **lines** --Specifies a print line count • **plcp** --Displays Physical Layer Control Protocol (PLCP) packets • **print** --Enables RF monitor printing mode • **probe** --Displays probe packets • **store** --Enables RF monitor storage mode

print	Enables packet printing. Use these options to turn on packet printing: • **hex** --Prints entire packets without formatting • **if** --Prints the in and out interfaces for packets • **iv** --Prints the packet Wired Equivalent Privacy (WEP) IV • **lines** --Prints the line count for the trace • **mic** --Prints the Cisco Message Integrity Check (MIC) • **plcp** --Displays the PLCP • **printf** --Prints using printf instead of buginf • **raw** --Prints without formatting data • **shortadr** --Prints MAC addresses in short form
stop-on-failure	Configures the access point to not restart when the radio driver fails.
trace	Enables trace mode. Use these options to turn on trace modes: • **off** --Turns off traces • **print** --Enables trace printing • **store** --Enables trace storage

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(4)JA	This command was introduced.
12.4(2)T	This command was integrated into Cisco IOS Release 12.4(2)T.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to display debugging information about radio options.

Examples

This example shows how to begin monitoring of all packets with CRC errors:

```
Router# debug dot11 dot11radio 0 monitor crc
```

Related Commands

Command	Description
debug dot11	Enables debugging of radio functions.
debug dot11 aaa	Enables debugging of dot11 AAA operations.

debug dot11 ids

Use the **debugdot11idseap** privileged EXEC command to enable debugging for wireless IDS monitoring. Use the **no** form of the command to disable IDS debugging.

[no] debug dot11 ids {**eap**| **cipher-errors**}

Syntax Description

eap	Activates debugging of IDS authentication events
cipher-errors	Activates debugging of cipher errors detected by IDS

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(4)JA	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

This example shows how to activate wireless IDS debugging for authentication events:

```
SOAP-AP# debug dot11 ids eap
```

Related Comma

Note This command is not supported on 1400 series bridges.

Command	Description
dot11 ids eap attempts	Configures limits on authentication attempts and EAPOL flooding on scanner access points in monitor mode
show debugging	Displays all debug settings and the debug packet headers

Command	Description
show dot11 ids eap	Displays wireless IDS statistics

debug dot11 ids mfp

Use the debug dot11 ids mfp privileged EXEC command to debug Management Frame Protection (MFP) operations on the access point.

{**[no] debug dot11 ids mfpap [all] [detectors] [events] [generators] [io] [reporting]| wds [all] [detectors] [events] [generators] [reporting] [statistics]| wlccp**}

Syntax Description

ap	Debugs MFP events on the access point.
all	Debugs all MFP events.
detectors	Debugs MFP detector key management events.
events	Debugs high level MFP events.
generators	Debugs MFP generator key management events.
io	**Debugs MFP IO (generate or detect frame) events.**
reporting	Debugs MFP reporting events.
statistics	Debugs MFP WDS statistics received from the detectors.
wds	Debugs MFP WDS events.
wlccp	Debugs MFP WLCCP messages.

Command Default

There are no defaults for this command.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(8)JA	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

This example shows how to debug the MFP detectors on the access point:

```
ap(config)# debug dot11 ids mfp ap detectors
```

Related Commands

Command	Description
dot11 ids mfp	Configures MFP parameters on the access point.
show dot11 ids mfp	Displays MFP parameters on the access point.

debug dot1x

To display 802.1X debugging information, use the **debugdot1x** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dot1x [**all**| **errors**| **events**| **feature**| **packets**| **redundancy**| **registry**| **state-machine**]

no debug dot1x [**all**| **errors**| **events**| **feature**| **packets**| **redundancy**| **registry**| **state-machine**]

Syntax Description

all	(Optional) Enables all 802.1X debugging messages.
errors	(Optional) Provides information about all 802.1X errors.
events	(Optional) Provides information about all 802.1X events.
feature	(Optional) Provides information about 802.1X features for switches only.
packets	(Optional) Provides information about all 802.1X packets.
redundancy	(Optional) Provides information about 802.1X redundancy.
registry	(Optional) Provides information about 802.1X registries.
state-machine	(Optional) Provides information regarding the 802.1X state machine.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(11)AX	This command was introduced.
12.1(14)EA1	The **authsm**, **backend**, **besm**, **core**, and **reauthsm** keywords were removed. The **errors**, **events**, **packets**, **registry**, and **state-machine** keywords were added.
12.3(2)XA	This command was integrated into Cisco IOS Release 12.3(2)XA.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Release	Modification
12.3(11)T	The **supplicant** keyword was added.
12.2(25)SEE	The **feature** keyword was added for switches only.
12.4(6)T	The **redundancy** keyword was added. The **aaa**, **process**, **rxdata**, **supplicant**, **txdata**, and **vlan** keywords were deleted.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output for the **debugdot1x** command:

```
Router# debug dot1x

Router-871#debug dot1x all
*Nov  7 13:07:56.872: dot1x-ev:dot1x_mgr_pre_process_eapol_pak: Role determination not
required on FastEthernet1.
*Nov  7 13:07:56.876: dot1x-packet:dot1x_mgr_process_eapol_pak: queuing an EAPOL pkt on
Authenticator Q
*Nov  7 13:07:56.876: dot1x-ev:Enqueued the eapol packet to the global authenticator queue
*Nov  7 13:07:56.876: dot1x-packet:Received an EAPOL frame on interface FastEthernet1
*Nov  7 13:07:56.876: dot1x-ev:Received pkt saddr =000f.23c4.a401 , daddr = 0180.c200.0003,

                    pae-ether-type = 888e.0202.0000
*Nov  7 13:07:56.876: dot1x-packet:Received an EAPOL-Logoff packet on interface FastEthernet1
*Nov  7 13:07:56.876: EAPOL pak dump rx
*Nov  7 13:07:56.876: EAPOL Version: 0x2  type: 0x2  length: 0x0000
*Nov  7 13:07:56.876: dot1x-sm:Posting EAPOL_LOGOFF on Client=82AC85CC
*Nov  7 13:07:56.876:     dot1x_auth Fa1: during state auth_authenticating, got event
7(eapolLogoff)
```

The fields in the output are self-explanatory.

Related Commands

Command	Description
clear dot1x	Clears 802.1X interface information.
identity profile default	Creates an identity profile and enters identity profile configuration mode.
show dot1x	Displays details for an identity profile.

debug dot1x (EtherSwitch)

To enable debugging of the 802.1x protocol when an Ethernet switch network module is installed, use the **debugdot1x** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dot1x {**all**| **authsm**| **backend**| **besm**| **core**| **reauthsm**}

no debug dot1x {**all**| **authsm**| **backend**| **besm**| **core**| **reauthsm**}

Syntax Description

all	Enables debugging of all conditions.
authsm	Enables debugging of the authenticator state machine, which is responsible for controlling access to the network through 802.1x-enabled ports.
backend	Enables debugging of the interaction between the 802.1x process and the router RADIUS client.
besm	Enables debugging of the backend state machine, which is responsible for relaying authentication request between the client and the authentication server.
core	Enables debugging of the 802.1x process, which includes 802.1x initialization, configuration, and the interaction with the port manager module.
reauthsm	Enables debugging of the reauthentication state machine, which manages periodic reauthentication of the client.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Release	Modification
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The **undebugdot1x** command is the same as the **nodebugdot1x** command.

Related Commands

Command	Description
show debugging	Displays information about the types of debugging that are enabled.
show dot1x	Displays 802.1x statistics, administrative status, and operational status for the router or for the specified interface.

debug drip event

To display debugging messages for Duplicate Ring Protocol (DRiP) events, use the **debugdripevent**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug drip event

no debug drip event

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is disabled for DRiP events.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(4)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

When a TrBRF interface is configured on the Remote Switch Module (RSM), the DRiP protocol is activated. The DRiP protocol adds the VLAN ID specified in the router command to its database and recognizes the VLAN as a locally configured, active VLAN.

Examples

The following is sample output from the **debugdripevent** command:

```
Router# debug drip event
```

DRiP gets a packet from the network:

```
612B92C0: 01000C00 00000000 0C501900 0000AAAA  .........P....**
612B92D0: 0300000C 00020000 00000100 0CCCCCCC  .............LLL
612B92E0: 00000C50 19000020 AAAA0300 000C0102  ...P... **......
612B92F0: 01010114 00000002 00000002 00000C50  ...............P
612B9300: 19000001 04C00064 04                 .....@.d.
```

DRiP gets a packet from the network:

```
Recvd. pak
```

DRiP recognizes that the VLAN ID it is getting is a new one from the network:

```
6116C840:                     0100 0CCCCCCC            ...LLL
6116C850: 00102F72 CBFB0024 AAAA0300 000C0102  ../rK{.$**......
6116C860: 01FF0214 0002E254 00015003 00102F72  ......bT..P.../r
6116C870: C8000010 04C00014 044003EB 14       H....@...@.k.
DRIP : remote update - Never heard of this vlan
```

DRiP attempts to resolve any conflicts when it discovers a new VLAN. The value action = 1 means to notify the local platform of change in state.

```
DRIP : resolve remote for vlan 20 in VLAN0
DRIP : resolve remote - action = 1
```

The local platform is notified of change in state:

```
DRIP Change notification active vlan 20
```

Another new VLAN ID was received in the packet:

```
DRIP : resolve remote for vlan 1003 in Vlan0
```

No action is required:

```
DRIP : resolve remote - action = 0
```

Thirty seconds have expired, and DRiP sends its local database entries to all its trunk ports:

```
DRIP : local timer expired
DRIP : transmit on 0000.0c50.1900, length = 24
612B92C0: 01000C00 00000000 0C501900 0000AAAA  .........P....**
612B92D0: 0300000C 00020000 00000100 0CCCCCCC  .............LLL
612B92E0: 00000C50 19000020 AAAA0300 000C0102  ...P... **......
612B92F0: 01FF0114 00000003 00000002 00000C50  ...............P
612B9300: 19000001 04C00064 04                 .....@.d.
```

debug drip packet

To display debugging messages for Duplicate Ring Protocol (DRiP) packets, use the **debugdrippacket**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug drip packet

no debug drip packet

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled for DRiP packets.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(4)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Before you use this command, you can optionally use the **cleardrip** command first. As a result the DRiP counters are reset to 0. If the DRiP counters begin to increment, the router is receiving packets.

Examples

The following is sample output from the **debugdrippacket** command:

```
Router# debug drip packet
```

The following type of output is displayed when a packet is entering the router and you use the **showdebug** command:

```
039E5FC0:     0100 0CCCCCCC 00E0A39B 3FFB0028    ...LLL.`#.?{.(
039E5FD0: AAAA0300 000C0102 01FF0314 0000A5F6  **............%v
039E5FE0: 00008805 00E0A39B 3C000000 04C00028  .....`#.<....@.(
039E5FF0: 04C00032 044003EB 0F                 .@.2.@.k.
039FBD20:                   01000C00 00000010          ........
```

The following type of output is displayed when a packet is sent by the router:

```
039FBD30: A6AEB450 0000AAAA 0300000C 00020000  &.4P..**........
039FBD40: 00000100 0CCCCCCC 0010A6AE B4500020  .....LLL..&.4P.
039FBD50: AAAA0300 000C0102 01FF0114 00000003  **..............
039FBD60: 00000002 0010A6AE B4500001 04C00064  ......&.4P...@.d
039FBD70: 04                                   .
```

Related Commands

Command	Description
debug drip event	Displays debugging messages for DRiP events.

debug dsc clock

To display debugging output for the time-division multiplexing (TDM) clock-switching events on the dial shelf controller (DSC), use the **debugdscclock** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

[execute-on] debug dsc clock

[execute-on] no debug dsc clock

Syntax Description

This command has no arguments or keywords; however, it can be used with the **execute-on** command.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(2)AA	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

To perform this command from the router shelf on the Cisco AS5800 series platform, use the **execute-onslot***slot-number***debugdscclock** form of this command.

The **debugdscclock** command displays TDM clock-switching events on the dial shelf controller. The information displayed includes the following:

- Clock configuration messages received from trunks via NBUS
- Dial shelf controller clock configuration messages from the router shelf over the dial shelf interface link
- Clock switchover algorithm events

Examples

The following example shows that the **debugdscclock** command has been enabled, and that trunk messages are received, and that the configuration message has been received:

```
AS5800# debug dsc clock
Dial Shelf Controller Clock debugging is on
AS5800#
00:02:55: Clock Addition msg of len 12 priority 8 from slot 1 port 1 on line 0
00:02:55: Trunk 1 has reloaded
```

Related Commands

Command	Description
execute-on	Executes commands remotely on a line card.

Command	Description
show dsc clock	Displays information about the dial shelf controller clock.

debug dsip

To display debugging output for Distributed System Interconnect Protocol (DSIP) used between a router shelf and a dial shelf, use the**debugdsip** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dsip {**all**| **api**| **boot**| **console**| **trace**| **transport**}

no debug dsip {**all**| **api**| **boot**| **console**| **trace**| **transport**}

Syntax Description

all	View all DSIP debugging messages.
api	View DSIP client interface (API) debugging messages.
boot	View DSIP booting messages that are generated when a download of the feature board image is occurring properly.
console	View DSIP console operation while debugging.
trace	Enable logging of header information concerning DSIP packets entering the system into a trace buffer. This logged information can be viewed with the **showdsiptracing** command.
transport	Debug the DSIP transport layer, the module that interacts with the underlying physical media driver.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(2)AA	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The **debugdsip** command is used to enable the display of debugging messages for DSIP between the router shelf and the dial shelf. Using this command, you can display booting messages generated when the download of an image occurs, view console operation, and trace logging of MAC header information and DSIP transport layer information as modules interact with the underlying physical media driver. This command can be applied to a single modem or a group of modems.

Once the **debugdsiptrace** command has been enabled, you can read the information captured in the trace buffer using the**showdsiptracing** command.

Examples

The following example indicates the **debugdsiptrace** command logs MAC headers of the various classes of DSIP packets. To view the logged information, use the **showdsiptracing** command:

```
AS5800# debug dsip trace
NIP tracing debugging is on
AS5800# show dsip tracing
NIP Control Packet Trace
------------------------------------------------------------
Dest:00e0.b093.2238 Src:0007.4c72.0058 Type:200B SrcShelf:1 SrcSlot:11
MsgType:0 MsgLen:82 Timestamp: 00:49:14
------------------------------------------------------------
Dest:00e0.b093.2238 Src:0007.4c72.0028 Type:200B SrcShelf:1 SrcSlot:5
MsgType:0 MsgLen:82 Timestamp: 00:49:14
------------------------------------------------------------
```

Related Commands

Command	Description
debug modem	Displays information about the dial shelf, including clocking information.
show dsip tracing	Displays DSIP media header information logged using the**debugdsiptrace** command.

debug dspapi

Note Effective with release 12.3(8)T, the **debugdspapi**command is replaced by the **debugvoipdspapi**command. See the **debugvoipdspapi**command for more information.

To enable debugging for Digital Signal Processor (DSP) application programming interface (API) message events, use the **debugdspapi** command in privileged EXEC mode. To reset the default value for this feature, use the **no** form of this command.

debug dspapi {**all**| **command**| **detail**| **error**| **notification**| **response**}

no debug dspapi {**all**| **command**| **detail**| **error**| **notification**| **response**}

Syntax Description

all	Enables all **debugdspapi**options (command, detail, error, notification and response).
command	Displays commands sent to the DSPs.
detail	Displays additional detail for the DSP API debugs enabled.
error	Displays any DSP API errors.
notification	Displays notification messages sent from the DSP (for example, tone detection notification).
response	Displays responses sent by the DSP (for example, responses to statistic requests).

Command Default

This command is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)XM	This command was introduced on the Cisco AS5300 and Cisco AS5800.
12.1(5)XM1	This command was implemented on the Cisco AS5350 and Cisco AS5400.
12.2(2)T	This command was implemented on the Cisco 1700, Cisco 2600 series, Cisco 3600 series, and the Cisco 3810.

Release	Modification
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.3(8)T	This command was replaced by the **debugvoipdspapi**command.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

DSP API message events used to communicate with DSPs are intended for use with Connexant (Nextport) and Texas Instrument (54x) DSPs. This command severely impacts performance and should be used only for single-call debug capture.

Examples

The following example shows how to enable debugging for all DSP API message events:

```
Router# debug dspapi all
```

Related Commands

Command	Description
debug hpi	Enables debugging for HPI message events.

debug dspfarm

To display digital signal processor (DSP) farm service debugging information, use the **debugdspfarm**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dspfarm {**all**| **errors**| **events**| **packets**}

no debug dspfarm

Syntax Description

all	All DSP-farm debug-trace information.
errors	DSP-farm errors.
events	DSP-farm events.
packets	DSP-farm packets.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YH	This command was introduced on the Cisco VG200.
12.2(13)T	This command was implemented on the Cisco 2600 series, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 3700 series.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The router on which this command is used must be equipped with one or more digital T1/E1 packet voice trunk network modules (NM-HDVs) or high-density voice (HDV) transcoding/conferencing DSP farms (NM-HDV-FARMs) to provide DSP resources.

Debugging is turned on for all DSP-farm-service sessions. You can debug multiple sessions simultaneously, with different levels of debugging for each.

Examples

The following is sample output from the **debugdspfarmevents** command:

```
Router# debug dspfarm events
DSP Farm service events debugging is on
*Mar  1 00:45:51: Sent 180 bytes to DSP 4 channel 2
*Mar  1 00:45:53: Sent 180 bytes to DSP 4 channel 3
```

```
*Mar  1 00:45:55: Sent 180 bytes to DSP 4 channel 1
*Mar  1 00:45:56: Sent 180 bytes to DSP 4 channel 2
*Mar  1 00:45:58: Sent 180 bytes to DSP 4 channel 3
*Mar  1 00:46:00: Sent 180 bytes to DSP 4 channel 1
*Mar  1 00:46:01: xapi_dspfarm_modify_connection: sess_id 26, conn_id 2705, conn_mode 3,
ripaddr 10.10.1.7, rport 20170
*Mar  1 00:46:01: dspfarm_process_appl_event_queue: XAPP eve 6311C4B0 rcvd
*Mar  1 00:46:01: dspfarm_find_stream: stream 63121F1C, found in sess 631143CC, cid 2705
*Mar  1 00:46:01: dspfarm_modify_connection: old_mode 4, new_mode 3
*Mar  1 00:46:01: dspfarm_close_local_rtp: stream 63121F1C, local_rtp_port 22656
*Mar  1 00:46:01: xapi_dspfarm_enqueue_event_to_appl: handle 63120634, event 6311C4C8,
eve_id 5, context 6311426C, result 0
*Mar  1 00:46:01: xapi_dspfarm_delete_connection: sess_id 26, conn_id 2705
*Mar  1 00:46:01: dspfarm_process_appl_event_queue: XAPP eve 6311C4E0 rcvd
*Mar  1 00:46:01: dspfarm_find_stream: stream 63121F1C, found in sess 631143CC, cid 2705
*Mar  1 00:46:01: dspfarm_close_local_rtp: stream 63121F1C, local_rtp_port 0
*Mar  1 00:46:01: dspfarm_release_dsp_resource: sess 631143CC, stream 63121F1C, num_stream
 3, sess_type 2, sess_dsp_id 2040000, stream_dsp_id 2040002
*Mar  1 00:46:01: dspfarm_drop_conference:slot 2 dsp 4 ch 2
*Mar  1 00:46:01: dspfarm_send_drop_conf: Sent drop_conference to DSP 4 ch 2
*Mar  1 00:46:01: dspfarm_xapp_enq: Sent msg 8 to DSPFARM
*Mar  1 00:46:01: xapi_dspfarm_enqueue_event_to_appl: handle 63120634, event 6311C4F8,
eve_id 9, context 6311426C, result 0
*Mar  1 00:46:01: dspfarm_process_dsp_event_queue: DSP eve 6312078C rcvd
*Mar  1 00:46:01: dspfarm_delete_stream: sess_id 26, conn_id 2705, stream 63121F1C,  in
sess 631143CC is freed
*Mar  1 00:46:01: Sent 180 bytes to DSP 4 channel 3
*Mar  1 00:46:04: Sent 180 bytes to DSP 4 channel 3
*Mar  1 00:46:05: xapi_dspfarm_modify_connection: sess_id 26, conn_id 2689, conn_mode 3,
ripaddr 10.10.1.5, rport 19514
*Mar  1 00:46:05: dspfarm_process_appl_event_queue: XAPP eve 6311C510 rcvd
*Mar  1 00:46:05: dspfarm_find_stream: stream 63121E34, found in sess 631143CC, cid 2689
*Mar  1 00:46:05: dspfarm_modify_connection: old_mode 4, new_mode 3
*Mar  1 00:46:05: dspfarm_close_local_rtp: stream 63121E34, local_rtp_port 25834
*Mar  1 00:46:05: xapi_dspfarm_enqueue_event_to_appl: handle 63120634, event 6311C528,
eve_id 5, context 63114244, result 0
*Mar  1 00:46:05: xapi_dspfarm_delete_connection: sess_id 26, conn_id 2689
*Mar  1 00:46:05: dspfarm_process_appl_event_queue: XAPP eve 6311C540 rcvd
*Mar  1 00:46:05: dspfarm_find_stream: stream 63121E34, found in sess 631143CC, cid 2689
*Mar  1 00:46:05: dspfarm_close_local_rtp: stream 63121E34, local_rtp_port 0
*Mar  1 00:46:05: dspfarm_release_dsp_resource: sess 631143CC, stream 63121E34, num_stream
 2, sess_type 2, sess_dsp_id 2040000, stream_dsp_id 2040001
*Mar  1 00:46:05: dspfarm_drop_conference:slot 2 dsp 4 ch 1
*Mar  1 00:46:05: dspfarm_send_drop_conf: Sent drop_conference to DSP 4 ch 1
*Mar  1 00:46:05: dspfarm_xapp_enq: Sent msg 8 to DSPFARM
*Mar  1 00:46:05: xapi_dspfarm_enqueue_event_to_appl: handle 63120634, event 6311C558,
eve_id 9, context 63114244, result 0
*Mar  1 00:46:05: dspfarm_process_dsp_event_queue: DSP eve 6311586C rcvd
*Mar  1 00:46:05: dspfarm_delete_stream: sess_id 26, conn_id 2689, stream 63121E34,  in
sess 631143CC is freed
*Mar  1 00:46:05: xapi_dspfarm_modify_connection: sess_id 26, conn_id 2721, conn_mode 3,
ripaddr 10.10.1.6, rport 21506
*Mar  1 00:46:05: dspfarm_process_appl_event_queue: XAPP eve 6311C570 rcvd
*Mar  1 00:46:05: dspfarm_find_stream: stream 63122004, found in sess 631143CC, cid 2721
*Mar  1 00:46:05: dspfarm_modify_connection: old_mode 4, new_mode 3
*Mar  1 00:46:05: dspfarm_close_local_rtp: stream 63122004, local_rtp_port 19912
*Mar  1 00:46:05: xapi_dspfarm_enqueue_event_to_appl: handle 63120634, event 6311C588,
eve_id 5, context 63114294, result 0
*Mar  1 00:46:05: xapi_dspfarm_delete_connection: sess_id 26, conn_id 2721
*Mar  1 00:46:05: dspfarm_process_appl_event_queue: XAPP eve 6311C5A0 rcvd
*Mar  1 00:46:05: dspfarm_find_stream: stream 63122004, found in sess 631143CC, cid 2721
*Mar  1 00:46:05: dspfarm_close_local_rtp: stream 63122004, local_rtp_port 0
*Mar  1 00:46:05: dspfarm_release_dsp_resource: sess 631143CC, stream 63122004, num_stream
 1, sess_type 2, sess_dsp_id 2040000, stream_dsp_id 2040003
*Mar  1 00:46:05: dspfarm_drop_conference:slot 2 dsp 4 ch 3
*Mar  1 00:46:05: dspfarm_drop_conference: Last conferee - closing the conf session
*Mar  1 00:46:05: dspfarm_send_close_conf: Sent close_conference to DSP 4
*Mar  1 00:46:05: dspfarm_drop_conference: Removed the conf in dsp 4
*Mar  1 00:46:05: dspfarm_xapp_enq: Sent msg 8 to DSPFARM
*Mar  1 00:46:05: xapi_dspfarm_enqueue_event_to_appl: handle 63120634, event 6311C5B8,
eve_id 9, context 63114294, result 0
*Mar  1 00:46:05: dspfarm_process_dsp_event_queue: DSP eve 6311586C rcvd
```

```
*Mar  1 00:46:05: dspfarm_delete_stream: sess_id 26, conn_id 2721, stream 63122004,  in
sess 631143CC is freed
```

Related Commands

Command	Description
debug frame-relay vc-bundle	Sets debugging for SCCP and its applications at one of four levels.
dspfarm (DSP farm)	Enables DSP-farm service.
sccp	Enables SCCP and its associated transcoding and conferencing applications.
show dspfarm	Displays summary information about DSP resources.

debug dspu activation

To display information on downstream physical unit (DSPU) activation, use the **debugdspuactivation** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dspu activation [*name*]

no debug dspu activation [*name*]

Syntax Description

name	(Optional) The host or physical unit (PU) name designation.

Command Modes

Privileged EXEC

Usage Guidelines

The **debugdspuactivation** command displays all DSPU activation traffic. To restrict the output to a specific host or PU, include the host or PU *name* argument. You cannot turn off debugging output for an individual PU if that PU has not been named in the **debugdspuactivation** command.

Examples

The following is sample output from the **debugdspuactivation** command. Not all intermediate numbers are shown for the "activated" and "deactivated" logical unit (LU) address ranges.

```
Router# debug dspu activation
DSPU: LS HOST3745 connected
DSPU: PU HOST3745 activated
DSPU: LU HOST3745-2 activated
DSPU: LU HOST3745-3 activated
.
.
.
DSPU: LU HOST3745-253 activated
DSPU: LU HOST3745-254 activated
DSPU: LU HOST3745-2 deactivated
DSPU: LU HOST3745-3 deactivated
.
.
.
DSPU: LU HOST3745-253 deactivated
DSPU: LU HOST3745-254 deactivated
DSPU: LS HOST3745 disconnected
DSPU: PU HOST3745 deactivated
```

The table below describes the significant fields shown in the display.

Table 8: debug dspu activation Field Descriptions

Field	Description
DSPU	Downstream PU debugging message.
LS	Link station (LS) event triggered the message.

Field	Description
PU	PU event triggered the message.
LU	LU event triggered the message.
HOST3745	Host name or PU name.
HOST3745-253	Host name or PU name and the LU address, separated by a dash.
connected activated disconnected deactivated	Event that occurred to trigger the message.

Related Commands

Command	Description
debug dspu packet	Displays information on a DSPU packet.
debug dspu state	Displays information on DSPU FSM state changes.
debug dspu trace	Displays information on DSPU trace activity.

debug dspu packet

To display information on a downstream physical unit (DSPU) packet, use the **debugdspupacket**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dspu packet [*name*]

no debug dspu packet [*name*]

Syntax Description

name	(Optional) The host or PU name designation.

Command Modes

Privileged EXEC

Usage Guidelines

The **debugdspupacket** command displays all DSPU packet data flowing through the router. To restrict the output to a specific host or physical unit (PU), include the host or PU *name* argument. You cannot turn off debugging output for an individual PU if that PU has not been named in the debug dspu packet command.

Examples

The following is sample output from the**debugdspupacket** command:

```
Router# debug dspu packet
DSPU: Rx: PU HOST3745 data length 12 data:
    2D0003002BE16B80 000D0201
DSPU: Tx: PU HOST3745 data length 25 data:
    2D0000032BE1EB80 000D020100850000 000C060000010000 00
DSPU: Rx: PU HOST3745 data length 12 data:
    2D0004002BE26B80 000D0201
DSPU: Tx: PU HOST3745 data length 25 data:
    2D0000042BE2EB80 000D020100850000 000C060000010000 00
```

The table below describes the significant fields shown in the display.

Table 9: debug dspu packet Field Descriptions

Field	Description
DSPU: Rx:	Received frame (packet) from the remote PU to the router PU.
DSPU: Tx:	Transmitted frame (packet) from the router PU to the remote PU.
PU HOST3745	Host name or PU associated with the transmit or receive.
data length 12 data:	Number of bytes of data, followed by up to 128 bytes of displayed data.

Related Commands

Command	Description
debug drip event	Displays debugging messages for DRiP packets.
debug dspu state	Displays information on DSPU FSM state changes.
debug dspu trace	Displays information on DSPU trace activity.

debug dspu state

To display information on downstream physical unit (DSPU) finite state machine (FSM) state changes, use the **debugdspustate** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dspu state [*name*]

no debug dspu state [*name*]

Syntax Description

name	(Optional) The host or physical unit (PU) name designation.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debugdspustate** command to display only the FSM state changes. To see all FSM activity, use the debug **dsputrace**command. You cannot turn off debugging output for an individual PU if that PU has not been named in the **debugdspustate** command.

Examples

The following is sample output from the **debugdspustate** command. Not all intermediate numbers are shown for the "activated" and "deactivated" logical unit (LU) address ranges.

```
Router# debug dspu state
DSPU: LS HOST3745: input=StartLs, Reset -> PendConOut
DSPU: LS HOST3745: input=ReqOpn.Cnf, PendConOut -> Xid
DSPU: LS HOST3745: input=Connect.Ind, Xid -> ConnIn
DSPU: LS HOST3745: input=Connected.Ind, ConnIn -> Connected
DSPU: PU HOST3745: input=Actpu, Reset -> Active
DSPU: LU HOST3745-2: input=uActlu, Reset -> upLuActive
DSPU: LU HOST3745-3: input=uActlu, Reset -> upLuActive
.
.
.
DSPU: LU HOST3745-253: input=uActlu, Reset -> upLuActive
DSPU: LU HOST3745-254: input=uActlu, Reset -> upLuActive
DSPU: LS HOST3745: input=PuStopped, Connected -> PendDisc
DSPU: LS HOST3745: input=Disc.Cnf, PendDisc -> PendClose
DSPU: LS HOST3745: input=Close.Cnf, PendClose -> Reset
DSPU: PU HOST3745: input=T2ResetPu, Active -> Reset
DSPU: LU HOST3745-2: input=uStopLu, upLuActive -> Reset
DSPU: LU HOST3745-3: input=uStopLu, upLuActive -> Reset
.
.
.
DSPU: LU HOST3745-253: input=uStopLu, upLuActive -> Reset
DSPU: LU HOST3745-254: input=uStopLu, upLuActive -> Reset
```

The table below describes the significant fields shown in the display.

Table 10: debug dspu state Field Descriptions

Field	Description
DSPU	Downstream PU debug message.
LS	Link station (LS) event triggered the message.
PU	PU event triggered the message.
LU	LU event triggered the message.
HOST3745-253	Host name or PU name and LU address.
input=input,	Input received by the FSM.
previous-state, -> current-state	Previous state and current new state as seen by the FSM.

Related Commands

Command	Description
debug drip event	Displays debugging messages for DRiP packets.
debug drip packet	Displays information on DSPU packet.
debug dspu trace	Displays information on DSPU trace activity.

debug dspu trace

To display information on downstream physical unit (DSPU) trace activity, which includes all finite state machine (FSM) activity, use the **debugdsputrace** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dspu trace [*name*]

no debug dspu trace [*name*]

Syntax Description

name	(Optional) The host or physical unit (PU) name designation.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debugdsputrace** command to display all FSM state changes. To see FSM state changes only, use the **debugdspustate** command. You cannot turn off debugging output for an individual PU if that PU has not been named in the **debugdsputrace** command.

Examples

The following is sample output from the **debugdsputrace** command:

```
Router# debug dspu trace
DSPU: LS HOST3745 input = 0 ->(1,a1)
DSPU: LS HOST3745 input = 5 ->(5,a6)
DSPU: LS HOST3745 input = 7 ->(5,a9)
DSPU: LS HOST3745 input = 9 ->(5,a28)
DSPU: LU HOST3745-2 in:0 s:0->(2,a1)
DSPU: LS HOST3745 input = 19 ->(8,a20)
DSPU: LS HOST3745 input = 18 ->(8,a17)
DSPU: LU HOST3745-3 in:0 s:0->(2,a1)
DSPU: LS HOST3745 input = 19 ->(8,a20)
DSPU: LS HOST3745 input = 18 ->(8,a17)
DSPU: LU HOST3745-252 in:0 s:0->(2,a1)
DSPU: LS HOST3745 input = 19 ->(8,a20)
DSPU: LS HOST3745 input = 18 ->(8,a17)
DSPU: LU HOST3745-253 in:0 s:0->(2,a1)
DSPU: LS HOST3745 input = 19 ->(8,a20)
DSPU: LS HOST3745 input = 18 ->(8,a17)
DSPU: LU HOST3745-254 in:0 s:0->(2,a1)
DSPU: LS HOST3745 input = 19 ->(8,a20)
```

The table below describes significant fields shown in the output.

Table 11: debug dspu trace Field Descriptions

Field	Description
7:23:57	Time stamp.
DSPU	Downstream PU debug message.

Field	Description
LS	Link station (LS) event triggered the message.
PU	A PU event triggered the message.
LU	LU event triggered the message.
HOST3745-253	Host name or PU name and LU address.
in:*input*s:*state*->(*new-state*, *action*)	String describing the following: • *input* --LU FSM input • *state* --Current FSM state • *new-state* --New FSM state • *action* --FSM action
input=*input* -> (*new-state*,*action*)	String describing the following: • *input* --PU or LS FSM input • *new-state* --New PU or LS FSM state • *action* --PU or LS FSM action

Related Commands

Command	Description
debug drip event	Displays debugging messages for DRiP packets.
debug drip packet	Displays information on DSPU packet.
debug dspu state	Displays information on DSPU FSM state changes.

debug dss ipx event

To display debugging messages for route change events that affect Internetwork Packet Exchange (IPX) Multilayer Switching (MLS), use the **debugdssipxevent** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dss ipx event

no debug dss ipx event

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the**debugdssipxevent** command:

```
Router#

debug dss ipx event
DSS IPX events debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface vlan 22
Router(config-if)# ipx access-group 800 out
05:51:36:DSS-feature:dss_ipxcache_version():idb:NULL, reason:42,
prefix:0, mask:FFFFFFFF
05:51:36:DSS-feature:dss_ipx_access_group():idb:Vlan22
05:51:36:DSS-feature:dss_ipx_access_list()
05:51:36:DSS-base 05:51:33.834 dss_ipx_invalidate_interface Vl22
05:51:36:DSS-base 05:51:33.834 dss_set_ipx_flowmask_reg 2
05:51:36:%IPX mls flowmask transition from 1 to 2 due to new status of
simple IPX access list on interfaces
```

Related Commands

Command	Description
debug mls rp	Displays various MLS debugging elements.