

debug appn pc

To display debugging information on Advanced Peer-to-Peer Networking (APPN) Path Control (PC) component activity, use the **debug appn pc** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug appn pc

no debug appn pc

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Usage Guidelines The PC component is responsible for passing Message Units (MUs) between the Data Link Control (DLC) layer and other APPN components. PC implements transmission priority by passing higher priority MUs to the DLC before lower priority MUs.

Examples The following is sample output from the **debug appn pc** command. In this example an MU is received from the network:

```
Router# debug appn pc

Turned on event 040000FF
APPN: ----- PC-----PC Deq REMOTE msg variant_name 2251
APPN: --PC-- mu received to PC lpid: A80AEC
APPN: --PC-- mu received from p_cep_id: 67C6F8
APPN: ----- PC-----PC Deq LSA_IPS from DLC
APPN: --PCX dequeued a DATA.IND
APPN: --- PC processing DL_DATA.ind
APPN: --PC-- mu_error_checker with no error, calling frr
APPN: --PC-- calling frr for packet received on LFSID: 1 2 3
APPN: ----- PC-----PC is sending MU to SC A90396
APPN: ----- SC-----send mu: A90396, rpc: 0, nws: 7, rh.b1: 90
APPN: SC: Send mu.snf: 8, th.b0: 2E, rh.b1: 90, dcf: 8
```

[Table 15](#) describes the significant fields shown in the display.

Table 15 *debug appn pc* Field Descriptions

Field	Description
APPN	APPN debugging output.
PC	PC component output.
Deq REMOTE	Message was received from the network.
mu received	Message is an MU.

Table 15 *debug appn pc Field Descriptions (continued)*

Field	Description
DATA.IND	MU contains data.
sending MU	MU is session traffic for an ISR session. The MU is forwarded to the Session Connector component for routing.

Related Commands

Command	Description
debug appn all	Turns on all possible debugging messages for APPN.

debug appn ps

To display debugging information on Advanced Peer-to-Peer Networking (APPN) Presentation Services (PS) component activity, use the **debug appn ps** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug appn ps

no debug appn ps

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Usage Guidelines The PS component is responsible for managing the Transaction Programs (TPs) used by APPN. TPs are used for sending and receiving searches, receiving resource registration, and sending and receiving topology updates.

Examples The following is sample output from the **debug appn ps** command. In this example a CP capabilities exchange is in progress.

```
Router# debug appn ps

Turned on event 200000FF
APPN: ---- CCA --- CP_CAPABILITIES_TP has started
APPN: ---- CCA --- About to wait for Partner to send CP_CAP
APPN: ---- CCA --- Partner LU name: NETA.PATTY
APPN: ---- CCA --- Mode Name: CPSVCMG
APPN: ---- CCA --- CGID: 78
APPN: ---- CCA --- About to send cp_cp_session_act to SS
APPN: ---- CCA --- Waiting for cp_cp_session_act_rsp from SS
APPN: ---- CCA --- Received cp_cp_session_act_rsp from SS
APPN: ---- CCA --- About to send CP_CAP to partner
APPN: ---- CCA --- Send to partner completed with rc=0, 0
APPN: ---- RCA --- Allocating conversation
APPN: ---- RCA --- Sending CP_CAPABILITIES
APPN: ---- RCA --- Getting conversation attributes
APPN: ---- RCA --- Waiting for partner to send CP_CAPABILITIES
APPN: ---- RCA --- Normal processing complete with cgid = 82
APPN: ---- RCA --- Deallocating CP_Capabilities conversation
```

[Table 16](#) describes the significant fields shown in the display.

Table 16 *debug appn ps* Field Descriptions

Field	Description
APPN	APPN debugging output.
CCA	CP Capabilities TP output.
RCA	Receive CP Capabilities TP output.

Related Commands

Command	Description
debug appn all	Turns on all possible debugging messages for APPN.

debug appn scm

To display debugging information on Advanced Peer-to-Peer Networking (APPN) Session Connector Manager (SCM) component activity, use the **debug appn scm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug appn scm

no debug appn scm

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Usage Guidelines The SCM component is responsible for the activation and deactivation of the local resources that route an intermediate session through the router.

Examples The following is sample output from the **debug appn scm** command. In this example an intermediate session traffic is being routed.

```
Router# debug appn scm

Turned on event 020000FF
Router#
APPN: ----- SCM-----SCM Deq a MU
APPN: ----- SCM-----SCM send ISR_INIT to SSI
APPN: ----- SCM----- (i05) Enter compare_fqpcid()
APPN: ----- SCM-----Adding new session_info table entry. addr=A93160
APPN: ----- SCM-----SCM Deq ISR_CINIT message
APPN: ----- SCM----- (i05) Enter compare_fqpcid()
APPN: ----- SCM-----SCM sends ASSIGN_LFSID to ASM
APPN: ----- SCM-----SCM Rcvd sync ASSIGN_LFSID from ASM
APPN: ----- SCM-----SCM PQenq a MU to ASM
APPN: ----- SCM-----SCM Deq a MU
APPN: ----- SCM----- (i05) Enter compare_fqpcid()
APPN: ----- SCM-----SCM PQenq BIND rsp to ASM
```

[Table 17](#) describes the significant fields shown in the display.

Table 17 *debug appn scm Field Descriptions*

Field	Description
APPN	APPN debugging output.
SCM	SCM component output.

Related Commands	Command	Description
	debug appn all	Turns on all possible debugging messages for APPN.

debug appn ss

To display session services (SS) events, use the **debug appn ss** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug appn ss

no debug appn ss

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The SS component generates unique session identifiers, activates and deactivates control point-to-control point (CP-CP) sessions, and assists logical units (LUs) in initiating and activating LU-LU sessions.

Examples

The following is sample output from the **debug appn ss** command. In this example CP-CP sessions between the router and another node are being activated.

```
Router# debug appn ss

Turned on event 100000FF
APPN: ----- SS ----- Deq ADJACENT_CP_CONTACTED message
APPN: ----- SS ----- Deq SESSST_SIGNAL message
APPN: ----- SS ----- Deq CP_CP_SESSION_ACT message
APPN: Sending ADJACENT_NN_1015 to SCM, adj_node_p=A6B980,cp_name=NETA.PATTY
APPN: ----- SS ----- Sending REQUEST_LAST_FRSN message to TRS
APPN: ----- SS ----- Receiving REQUEST_LAST_FRSN_RSP from TRS
APPN: ----- SS ----- Sending ACTIVE_CP_STATUS CONLOSER message to DS
APPN: ----- SS ----- Sending ACTIVE_CP_STATUS CONLOSER message to MS
APPN: ----- SS ----- Sending ACTIVE_CP_STATUS CONLOSER message to TRS
APPN: ----- SS ----- Sending CP_CP_SESSION_ACT_RSP message to CCA TP
APPN: ----- SS ----- Sending PENDING_ACTIVE_CP_STATUS CONWINNER message to DS
APPN: ----- SS ----- Sending REQUEST_LAST_FRSN message to TRS
APPN: ----- SS ----- Receiving REQUEST_LAST_FRSN_RSP from TRS
APPN: ----- SS ----- Sending ACT_CP_CP_SESSION message to RCA TP
APPN: ----- SS ----- Deq ASSIGN_PCID message
APPN: ----- SS ----- Sending ASSIGN_PCID_RSP message to someone
APPN: ----- SS ----- Deq INIT_SIGNAL message
APPN: ----- SS ----- Sending REQUEST_COS_TPF_VECTOR message to TRS
APPN: ----- SS ----- Receiving an REQUEST_COS_TPF_VECTOR_RSP from TRS
APPN: ----- SS ----- Sending REQUEST_SINGLE_HOP_ROUTE message to TRS
APPN: ----- SS ----- Receiving an REQUEST_SINGLE_HOP_ROUTE_RSP from TRS
APPN: ----- SS ----- Sending ACTIVATE_ROUTE message to CS
APPN: ----- SS ----- Deq ACTIVATE_ROUTE_RSP message
APPN: ----- SS ----- Sending CINIT_SIGNAL message to SM
APPN: ----- SS ----- Deq ACT_CP_CP_SESSION_RSP message
APPN: -- SS-----SS ssp00, act_cp_cp_session_rsp received, sense_code=0, cgid=5C,
ips@=A93790
APPN: Sending ADJACENT_NN_1015 to SCM, adj_node_p=A6B980,cp_name=18s
```

```

APPN: ----- SS ----- Sending ACTIVE CP_STATUS CONWINNER message to DS
APPN: ----- SS ----- Sending ACTIVE CP_STATUS CONWINNER message to MS
APPN: ----- SS ----- Sending ACTIVE CP_STATUS CONWINNER message to TRS

```

[Table 18](#) describes the significant fields shown in the display.

Table 18 *debug appn ss Field Descriptions*

Field	Description
APPN	APPN debugging output.
SS	SS component output.

Related Commands

Command	Description
debug appn all	Turns on all possible debugging messages for APPN.

debug appn trs

To display debugging information on Advanced Peer-to-Peer Networking (APPN) Topology and Routing Services (TRS) component activity, use the **debug appn trs** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug appn trs

no debug appn trs

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The TRS component is responsible for creating and maintaining the topology database, creating and maintaining the class of service database, and computing and caching optimal routes through the network.

Examples

The following is sample output from the **debug appn trs** command:

```
Router# debug appn trs

Turned on event 400000FF
APPN: ----- TRS ----- Received a QUERY_CPNAME
APPN: ----- TRS ----- Received a REQUEST_ROUTE
APPN: ----- TRS ----- check_node node_name=NETA.LISA
APPN: ----- TRS ----- check_node node_index=0
APPN: ----- TRS ----- check_node node_weight=60
APPN: ----- TRS ----- add index 484 to origin description list
APPN: ----- TRS ----- add index 0 to dest description list
APPN: ----- TRS ----- origin tg_vector is NULL
APPN: ----- TRS ----- weight_to_origin = 0
APPN: ----- TRS ----- weight_to_dest = 0
APPN: ----- TRS ----- u_b_s_f weight = 30
APPN: ----- TRS ----- u_b_s_f prev_weight = 2147483647
APPN: ----- TRS ----- u_b_s_f origin_index = 484
APPN: ----- TRS ----- u_b_s_f dest_index = 0
APPN: ----- TRS ----- b_r_s_f weight = 30
APPN: ----- TRS ----- b_r_s_f origin_index = 484
APPN: ----- TRS ----- b_r_s_f dest_index = 0
APPN: ----- TRS ----- Received a REQUEST_ROUTE
APPN: ----- TRS ----- check_node node_name=NETA.LISA
APPN: ----- TRS ----- check_node node_index=0
APPN: ----- TRS ----- check_node node_weight=60
APPN: ----- TRS ----- check_node node_name=NETA.BART
APPN: ----- TRS ----- check_node node_index=484
APPN: ----- TRS ----- check_node node_weight=60
APPN: ----- TRS ----- add index 484 to origin description list
APPN: ----- TRS ----- add index 0 to dest description list
APPN: ----- TRS ----- origin_tg_weight to non-VN=30
APPN: ----- TRS ----- origin_node_weight to non-VN=60
APPN: ----- TRS ----- weight_to_origin = 90
APPN: ----- TRS ----- weight_to_dest = 0
```

```

APPN: ----- TRS ----- u_b_s_f weight = 120
APPN: ----- TRS ----- u_b_s_f prev_weight = 2147483647
APPN: ----- TRS ----- u_b_s_f origin_index = 484
APPN: ----- TRS ----- u_b_s_f dest_index = 0
APPN: ----- TRS ----- b_r_s_f weight = 120
APPN: ----- TRS ----- b_r_s_f origin_index = 484
APPN: ----- TRS ----- b_r_s_f dest_index = 0

```

[Table 19](#) describes the significant fields shown in the display.

Table 19 *debug appn trs* Field Descriptions

Field	Description
APPN	APPN debugging output.
TRS	TRS component output.

debug arap

To display AppleTalk Remote Access Protocol (ARAP) events, use the **debug arap** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug arap {internal | memory | mnp4 | v42bis} [linenum [aux | console | tty | vty]]
```

```
no debug arap {internal | memory | mnp4 | v42bis} [linenum [aux | console | tty | vty]]
```

Syntax Description

internal	Debugs internal ARA packets.
memory	Debugs memory allocation for ARA.
mnp4	Debugs low-level asynchronous serial protocol.
v42bis	Debugs V.42bis compression.
<i>linenum</i>	(Optional) Line number. The number ranges from 0 to 999, depending on what type of line is selected.
aux	(Optional) Auxiliary line.
console	(Optional) Primary terminal line.
tty	(Optional) Physical terminal asynchronous line.
vty	(Optional) Virtual terminal line.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debug arap** command with the **debug callback** command on access servers to debug dialin and callback events.

Use the **debug modem** command to help catch problems related to ARAP autodetection (that is, **autoselect arap**). These problems are very common and are most often caused by modems, which are the most common cause of failure in ARAP connection and configuration sessions.

Examples

The following is sample output from the **debug arap internal** command:

```
Router# debug arap internal

ARAP: ----- SRVVERSION -----
ARAP: ----- ACKing 0 -----
ARAP: ----- AUTH_CHALLENGE -----
arapsec_local_account setting up callback
ARAP: ----- ACKing 1 -----
ARAP: ----- AUTH_RESPONSE -----
arap_startup initiating callback ARAP 2.0
ARAP: ----- CALLBACK -----
TTY7 Callback process initiated, user: dialback dialstring 40
TTY7 Callback forced wait = 4 seconds
TTY7 ARAP Callback Successful - await exec/autoselect pickup
TTY7: Callback in effect
ARAP: ----- STARTINFOFROMSERVER -----
ARAP: ----- ACKing 0 -----
ARAP: ----- ZONELISTINFO -----
```

■ debug arap

```
ARAP: ----- ZONELISTINFO -----  
ARAP: ----- ZONELISTINFO -----  
ARAP: ----- ZONELISTINFO -----  
ARAP: ----- ZONELISTINFO -----
```

Related Commands

Command	Description
debug callback	Displays callback events when the router is using a modem and a chat script to call back on a terminal line.
debug modem	Observes modem line activity on an access server.

debug arp

To display information on Address Resolution Protocol (ARP) transactions, use the **debug arp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug arp

no debug arp

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Use this command when some nodes on a TCP/IP network are responding, but others are not. It shows whether the router is sending ARP packets and whether it is receiving ARP packets.

Examples

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

```
Router# debug arp

IP ARP: sent req src 172.16.22.7 0000.0c01.e117, dst 172.16.22.96 0000.0000.0000
IP ARP: rcvd rep src 172.16.22.96 0800.2010.b908, dst 172.16.22.7
IP ARP: rcvd req src 172.16.6.10 0000.0c00.6fa2, dst 172.16.6.62
IP ARP: rep filtered src 172.16.22.7 aa92.1b36.a456, dst 255.255.255.255 ffff.ffff.ffff
IP ARP: rep filtered src 172.16.9.7 0000.0c00.6b31, dst 172.16.22.7 0800.2010.b908
```

In the output, each line of output represents an ARP packet that the router sent or received. Explanations for the individual lines of output follow.

The first line indicates that the router at IP address 172.16.22.7 and MAC address 0000.0c01.e117 sent an ARP request for the MAC address of the host at 172.16.22.96. The series of zeros (0000.0000.0000) following this address indicate that the router is currently unaware of the MAC address.

```
IP ARP: sent req src 172.16.22.7 0000.0c01.e117, dst 172.16.22.96 0000.0000.0000
```

The second line indicates that the router at IP address 172.16.22.7 receives a reply from the host at 172.16.22.96 indicating that its MAC address is 0800.2010.b908:

```
IP ARP: rcvd rep src 172.16.22.96 0800.2010.b908, dst 172.16.22.7
```

The third line indicates that the router receives an ARP request from the host at 172.16.6.10 requesting the MAC address for the host at 172.16.6.62:

```
IP ARP: rcvd req src 172.16.6.10 0000.0c00.6fa2, dst 172.16.6.62
```

The fourth line indicates that another host on the network attempted to send the router an ARP reply for its own address. The router ignores meaningless replies. Usually, meaningless replies happen if a bridge is being run in parallel with the router and is allowing ARP to be bridged. This condition indicates a network misconfiguration.

```
IP ARP: rep filtered src 172.16.22.7 aa92.1b36.a456, dst 255.255.255.255 ffff.ffff.ffff
```

The fifth line indicates that another host on the network attempted to inform the router that it is on network 172.16.9.7, but the router does not know that the network is attached to a different router interface. The remote host (probably a PC or an X terminal) is misconfigured. If the router were to install this entry, it would deny service to the real machine on the proper cable.

```
IP ARP: rep filtered src 172.16.9.7 0000.0c00.6b31, dst 172.16.22.7 0800.2010.b908
```

debug asp packet

To display information on all asynchronous security protocols operating on the router, use the **debug asp packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug asp packet

no debug asp packet

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The router uses asynchronous security protocols from companies including ADT Security Systems, Inc., Adplex, and Diebold to transport alarm blocks between two devices (such as a security alarm system console and an alarm panel). The alarm blocks are transported in pass-through mode using BSTUN encapsulation.

Examples

The following is partial sample output from the **debug asp packet** command for asynchronous security protocols when packet debugging is enabled on an asynchronous line carrying Diebold alarm traffic. In this example, two polls are sent from the Diebold alarm console to two alarm panels that are multidropped from a single EIA/TIA-232 interface. The alarm panels have device addresses F0 and F1. The example trace indicates that F1 is responding and F0 is not responding. At this point, you need to examine the physical link and possibly use a datascoper to determine why the device is not responding.

```
Router# debug asp packet

12:19:48: ASP: Serial5: ADI-Rx: Data (4 bytes): F1FF4C42
12:19:49: ASP: Serial5: ADI-Tx: Data (1 bytes): 88
12:19:49: ASP: Serial5: ADI-Rx: Data (4 bytes): F0FF9B94
12:20:47: ASP: Serial5: ADI-Rx: Data (4 bytes): F1FF757B
12:20:48: ASP: Serial5: ADI-Tx: Data (1 bytes): F3
12:20:48: ASP: Serial5: ADI-Rx: Data (4 bytes): F0FFB1BE
12:21:46: ASP: Serial5: ADI-Rx: Data (4 bytes): F1FFE6E8
12:21:46: ASP: Serial5: ADI-Tx: Data (1 bytes): 6F
12:21:46: ASP: Serial5: ADI-Rx: Data (4 bytes): F0FFC1CE
```

[Table 20](#) describes the significant fields shown in the display.

Table 20 *debug asp packet Field Descriptions*

Field	Description
ASP	Asynchronous security protocol packet.
Serial5	Interface receiving and sending the packet.
ADI-Rx	Packet is being received.
ADI-T	Packet is being sent.

Table 20 *debug asp packet Field Descriptions (continued)*

Field	Description
Data (<i>n</i> bytes)	Type and size of the packet.
F1FF4c42	Alarm panel device address.

debug async async-queue

To display debug messages for asynchronous rotary line queuing, use the **debug async async-queue** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug async async-queue

no debug async async-queue

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.1(1)T	This command was introduced.

Examples The following example starts the asynchronous rotary line queuing debugging display:

```
Router# debug async async-queue

*Mar 2 03:50:28.377: AsyncQ: First connection to be queued - starting the AsyncQ manager
*Mar 2 03:50:28.377: AsyncQ: Enabling the AsyncQ manager
*Mar 2 03:50:28.377: AsyncQ: Started the AsyncQ manager process with pid 98
*Mar 2 03:50:28.381: AsyncQ: Created a Waiting TTY on TTY66 with pid 99
*Mar 2 03:50:30.164: WaitingTTY66: Did Authentication on waiting TTY (VTY)
*Mar 2 03:50:30.168: AsyncQ: Received ASYNCQ_MSG_ADD
*Mar 2 03:50:30.168: AsyncQ: New queue, adding this connection as the first element
*Mar 2 03:50:34.920: AsyncQ: Created a Waiting TTY on TTY67 with pid 100
*Mar 2 03:50:36.783: WaitingTTY67: Did Authentication on waiting TTY (VTY)
*Mar 2 03:50:36.787: AsyncQ: Received ASYNCQ_MSG_ADD
*Mar 2 03:50:36.787: AsyncQ: Queue exists, adding this connection to the end of the queue
```

Related Commands	Command	Description
	debug ip tcp transactions	Enables the IP TCP transactions debugging display to observe significant transactions such as state changes, retransmissions, and duplicate packets.
	debug modem	Enables the modem debugging display to observe modem line activity on an access server.

debug atm bundle error

To display debug messages for switched virtual circuit (SVC) bundle errors, use the **debug atm bundle error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug atm bundle error

no debug atm bundle error

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

Examples The following example provides output for the **debug atm bundle error** command:

```
Router# debug atm bundle error
```

Related Commands	Command	Description
	debug atm bundle events	Displays SVC bundle events.

debug atm bundle events

To display switched virtual circuit (SVC) bundle events, use the **debug atm bundle events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug atm bundle events

no debug atm bundle events

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(4)T	This command was introduced.

Examples The following example provides output for the **debug atm bundle events** command:

```
Router# debug atm bundle events

01:14:35:BUNDLE EVENT(test):b_update_vc for four with bstate 1, vc_state4
01:14:35:BUNDLE EVENT(test):bmupdate active precedence 0x01 0
01:14:35:BUNDLE EVENT(test):bmupdate active precedence 0x02 0
01:14:35:BUNDLE EVENT(test):bmupdate active precedence 0x04 0
01:14:35:BUNDLE EVENT(test):bmupdate active precedence 0x08 0
01:14:35:BUNDLE EVENT(test):bmupdate active precedence 0x10 0
01:14:35:BUNDLE EVENT(test):bmupdate active precedence 0x20 0
01:14:35:BUNDLE EVENT(test):bmupdate active precedence 0x40 0
01:14:35:BUNDLE EVENT(test):bmupdate active precedence 0x80 0 -
01:14:35:BUNDLE EVENT(test):bundle precedence updated
```

[Table 21](#) describes the significant fields shown in the display.

Table 21 *debug atm events Field Description*

Field	Description
01:14:35	Local time on the router in hours:minutes:seconds.
BUNDLE EVENT(test)	Bundle event for bundle by that name.
b_update_vc for four with bstate 1, vc_state 1	Test describing the bundle event.

Related Commands	Command	Description
	debug atm bundle error	Displays debug messages for SVC bundle errors.

debug atm events

To display ATM events, use the **debug atm events** command in privileged EXEC mode. To disable event debugging output, use the **no** form of this command.

debug atm events

no debug atm events

Syntax Description This command has no arguments or keywords.

Defaults ATM event debugging is disabled.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.1(3)XJ	This command was introduced on the Cisco 1700 series routers.
	12.1(5)XR1	This command was implemented on the Cisco IAD2420 Series.
	12.2(4)T	This command was integrated into Cisco IOS 12.2(4)T.

Usage Guidelines This command displays ATM events that occur on the ATM interface processor and is useful for diagnosing problems in an ATM network. It provides an overall picture of the stability of the network. In a stable network, the **debug atm events** command does not return any information. If the command generates numerous messages, the messages can indicate the possible source of problems.

When configuring or making changes to a router or interface for ATM, enable the **debug atm events** command. Doing so alerts you to the progress of the changes or to any errors that might result. Also use this command periodically when you suspect network problems.

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

```
Router# debug atm events

RESET(ATM4/0): PLIM type is 1, Rate is 100Mbps
aip_disable(ATM4/0): state=1
config(ATM4/0)
aip_love_note(ATM4/0): asr=0x201
aip_enable(ATM4/0)
aip_love_note(ATM4/0): asr=0x4000
aip_enable(ATM4/0): restarting VCs: 7
aip_setup_vc(ATM4/0): vc:1 vpi:1 vci:1
aip_love_note(ATM4/0): asr=0x200
aip_setup_vc(ATM4/0): vc:2 vpi:2 vci:2
aip_love_note(ATM4/0): asr=0x200
aip_setup_vc(ATM4/0): vc:3 vpi:3 vci:3
aip_love_note(ATM4/0): asr=0x200
aip_setup_vc(ATM4/0): vc:4 vpi:4 vci:4
```

```

aip_love_note(ATM4/0): asr=0x200
aip_setup_vc(ATM4/0): vc:6 vpi:6 vci:6
aip_love_note(ATM4/0): asr=0x200
aip_setup_vc(ATM4/0): vc:7 vpi:7 vci:7
aip_love_note(ATM4/0): asr=0x200
aip_setup_vc(ATM4/0): vc:11 vpi:11 vci:11
aip_love_note(ATM4/0): asr=0x200

```

Table 22 describes the significant fields shown in the display.

Table 22 *debug atm events Field Descriptions*

Field	Description
PLIM type	Indicates the interface rate in megabits per second (Mbps). Possible values are: <ul style="list-style-type: none"> • 1 = TAXI(4B5B) 100 Mbps • 2 = SONET 155 Mbps • 3 = E3 34 Mbps
state	Indicates current state of the ATM Interface Processor (AIP). Possible values are: <ul style="list-style-type: none"> • 1 = An ENABLE will be issued soon. • 0 = The AIP will remain shut down.
asr	Defines a bitmask, which indicates actions or completions to commands. Valid bitmask values are: <ul style="list-style-type: none"> • 0x0800 = AIP crashed, reload may be required. • 0x0400 = AIP detected a carrier state change. • 0x0n00 = Command completion status. Command completion status codes are: <ul style="list-style-type: none"> - n = 8 Invalid Physical Layer Interface Module (PLIM) detected - n = 4 Command failed - n = 2 Command completed successfully - n = 1 CONFIG request failed - n = 0 Invalid value

The following line indicates that the AIP was reset. The PLIM TYPE detected was 1, so the maximum rate is set to 100 Mbps.

```
RESET(ATM4/0): PLIM type is 1, Rate is 100Mbps
```

The following line indicates that the AIP was given a shutdown command, but the current configuration indicates that the AIP should be up:

```
aip_disable(ATM4/0): state=1
```

The following line indicates that a configuration command has been completed by the AIP:

```
aip_love_note(ATM4/0): asr=0x201
```

The following line indicates that the AIP was given a no shutdown command to take it out of shutdown:

```
aip_enable(ATM4/0)
```

The following line indicates that the AIP detected a carrier state change. It does not indicate that the carrier is down or up, only that it has changed.

```
aip_love_note(ATM4/0): asr=0x4000
```

The following line of output indicates that the AIP enable function is restarting all permanent virtual circuits (PVCs) automatically:

```
aip_enable(ATM4/0): restarting VCs: 7
```

The following lines of output indicate that PVC 1 was set up and a successful completion code was returned:

```
aip_setup_vc(ATM4/0): vc:1 vpi:1 vci:1  
aip_love_note(ATM4/0): asr=0x200
```

debug atm native

To display ATM switched virtual circuit (SVC) signaling events, use the **debug atm native** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug atm native {[api] | [conn] | [error] | [filter]}
```

```
no debug atm native
```

Syntax Description

api	(Optional) Native ATM application programming interface (API). Displays events that occur as a result of the exchange between the native ATM API and the signaling API.
conn	(Optional) Native ATM connection manager. Displays internal connection manager events for the native ATM API.
error	(Optional) Native ATM error. Displays errors that occur during the setup of an ATM SVC.
filter	(Optional) Native ATM filter. Displays the internal network service access point (NSAP) filter events of the native ATM API.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

Native ATM API is the layer above the signaling API. Static map and Resource Reservation Protocol (RSVP) clients use the native ATM API to interact with the signaling API to create ATM SVCs.

Use the **debug atm native** command to diagnose problems in the creation of static map and RSVP ATM SVCs.

Examples

The following is sample output for the **debug atm native** command with the **api** keyword:

```
Router# debug atm native api

0:24:59:NATIVE ATM :associate endpoint
00:24:59:NATIVE ATM :ID (3) prep outgoing call, conn_type 0
00:24:59:NATIVE ATM :ID (3) set connection attribute for 5
00:24:59:NATIVE ATM :ID (3) query connection attribute 8
00:24:59:NATIVE ATM :ID (3) set connection attribute for 8
00:24:59:NATIVE ATM :ID (3) set connection attribute for 9
00:24:59:NATIVE ATM :ID (3) set connection attribute for 10
00:24:59:NATIVE ATM :ID (3) set connection attribute for 7
00:24:59:NATIVE ATM :ID (3) set connection attribute for 6
00:24:59:NATIVE ATM :ID (3) set connection attribute for 2
00:24:59:NATIVE ATM :ID (3) set connection attribute for 0
00:24:59:NATIVE ATM :ID (3) query connection attribute 12
00:24:59:NATIVE ATM :ID (3) set connection attribute for 12
00:24:59:NATIVE ATM :ID (3) query connection attribute 13
00:24:59:NATIVE ATM :ID (3) set connection attribute for 13
00:24:59:NATIVE ATM :ID (3) connect outgoing call
00:24:59:NATIVE ATM :ID (3) callback, CONNECT received
```

debug atm nbma

To display setup and teardown events for ATM switched virtual circuits (SVCs) configured using the Resource Reservation Protocol (RSVP), use the **debug atm nbma** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug atm nbma [api]

no debug atm nbma

Syntax Description

api	(Optional) Nonbroadcast multiaccess (NBMA) ATM application programming interface (API). Displays events that occur as a result of the exchange between RSVP and the NBMA API.
------------	---

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

Use the **debug atm nbma** command to diagnose problems in the creation of RSVP SVCs.

The RSVP application creates SVCs by using the NBMA API. The **debug atm nbma** command with the **api** keyword displays events that occur as a result of the exchange between RSVP and the NBMA API.

Examples

The following is sample output for the **debug atm nbma** command:

```
Router# debug atm nbma api

00:52:50:NBMA-ATM-API - atm_setup_req
00:52:50:NBMA_ATM-API - nbma_atm_fill_blli
00:52:50:NBMA_ATM-API - nbma_atm_fill_bhli
00:52:50:NBMA_ATM-API - nbma_atm_callbackMsg - NATIVE_ATM_OUTGOING_CALL_ACTIVE
00:52:50:NBMA_ATM-API - rcv_outgoing_call_active
00:52:50:NBMA_ATM-API - nbma_svc_lookup
```

debug atm oam cc

To display ATM operation, administration, and maintenance (OAM) F5 continuity check (CC) management activity, use the **debug atm oam cc** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug atm oam cc [*interface atm number*]

no debug atm oam cc [*interface atm number*]

Syntax Description

interface atm number (Optional) Number of the ATM interface.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XB	This command was introduced.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.

Examples

The following sample output for the **debug atm oam cc** command records activity beginning with the entry of the **oam-pvc manage cc** command and ending with the entry of the **no oam-pvc manage cc** command. The ATM 0 interface is specified, and the “both” segment direction is specified. The output shows an activation request sent and confirmed, a series of CC cells sent by the routers on each end of the segment, and a deactivation request and confirmation.

```
Router# debug atm oam cc interface atm0

Generic ATM:
  ATM OAM CC cells debugging is on
Router#
00:15:05: CC ACTIVATE MSG (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM
Type:8 OAM Func:1 Direction:3 CTag:5
00:15:05: CC ACTIVATE CONFIRM MSG (ATM0) O:VCD#1 VC 1/40 OAM Cell
Type:4 OAM Type:8 OAM Func:1 Direction:3 CTag:5
00:15:06: CC CELL (ATM0) O:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1
00:15:07: CC CELL (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:08: CC CELL (ATM0) O:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:09: CC CELL (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:10: CC CELL (ATM0) O:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:11: CC CELL (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:12: CC CELL (ATM0) O:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:13: CC CELL (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:14: CC CELL (ATM0) O:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:15: CC CELL (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:16: CC CELL (ATM0) O:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:17: CC CELL (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:18: CC CELL (ATM0) O:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
00:15:19: CC CELL (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM Type:1 OAM Func:4
```

```
00:15:19: CC DEACTIVATE MSG (ATM0) I:VCD#1 VC 1/40 OAM Cell Type:4 OAM
Type:8 OAM Func:1 Direction:3 CTag:6
00:15:19: CC DEACTIVATE CONFIRM MSG (ATM0) O:VCD#1 VC 1/40 OAM Cell
Type:4 OAM Type:8 OAM Func:1 Direction:3 CTag:6
```

Table 23 describes the significant fields shown in the display.

Table 23 *debug atm oam cc Field Descriptions*

Field	Description
00:15:05	Time stamp.
CC ACTIVATE MSG (ATM0)	Message type and interface.
0	Source.
1	Sink.
VC 1/40	Virtual circuit identifier.
Direction:3	Direction in which the cells are traveling. May be one of the following values: 1— local router is the sink. 2— local router is the source. 3— both routers operate as the source and sink.

Related Commands

Command	Description
oam-pvc manage cc	Configures ATM OAM F5 CC management.
show atm pvc	Displays all ATM PVCs and traffic information.

debug atm state

To display the states for Asynchronous Transfer Mode (ATM) common connections on the networking device, use the **debug atm state** command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug atm state

no debug atm state

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Release	Modification
	11.3	This command was introduced.

Examples The following example shuts the interface down and displays the debugging messages in regard to the ATM interface on the networking device:

```
Router# debug atm state

ATM VC States debugging is on

Router# show debug

Generic ATM:
  ATM VC States debugging is on

Router# configure terminal
Router(config)# interface atm 2/0.2
Router(config-if)# shutdown

*Aug  8 17:45:38.987: Changing vc 3/100vc-state to ATM_VC_SHUTTING_DOWN
*Aug  8 17:45:38.991: Changing vc 3/100vc-state to ATM_VC_NOT_IN_SERVICE
```

The following example turns the interface back on and displays the debugging messages in regard to the ATM interface on the networking device:

```
Router(config)# interface atm 2/0.2
Router(config-if)# no shutdown

*Aug  8 17:45:44.711: Changing vc 3/100 vc-state to ATM_VC_ESTABLISHING_VC
*Aug  8 17:45:44.711: Changing vc 3/100 vc-state to ATM_VC_NOT_VERIFIED
*Aug  8 17:45:44.711: Changing vc 3/100 vc-state to ATM_VC_UP
```

Related Commands

Command	Description
debug atm ha-state	Displays the ATM HA state on the networking device.

debug backhaul-session-manager session

To debug all the available sessions or a specified session, use the **debug backhaul-session-manager session** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug backhaul-session-manager session {state | xport} {all | session-id}

no debug backhaul-session-manager session {state | xport} {all | session-id}



Caution

Use caution when enabling this debug command in a live system. It produces significant amounts of output, which could lead to a disruption of service.

Syntax Description

state	Shows information about state transitions. Possible states are as follows: SESS_SET_IDLE: A session-set has been created. SESS_SET_OOS: A session(s) has been added to session-group(s). No ACTIVE notification has been received from Virtual Switch Controller (VSC). SESS_SET_ACTIVE_IS: An ACTIVE notification has been received over one in-service session-group. STANDBY notification has not been received on any available session-group(s). SESS_SET_STNDBY_IS: A STANDBY notification is received, but there is no in-service active session-group available. SESS_SET_FULL_IS: A session-group in-service that has ACTIVE notification and at least one session-group in-service that has STANDBY notification. SESS_SET_SWITCH_OVER: An ACTIVE notification is received on session-group in-service, which had received STANDBY notification.
xport	Provides traces for all packets (protocol data units (PDUs)), application PDUs, and also session-manager messages.
all	All available sessions.
<i>session-id</i>	A specified session.

Defaults

Debugging for backhaul-session-manager session is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.2(2)T	Support for this command was introduced on the Cisco 7200 series routers.
12.2(4)T	This command was implemented on the Cisco 2600 series, Cisco 3600 series, and Cisco MC3810.
12.2(2)XB	This command was implemented on the Cisco AS5350 and Cisco AS5400.
12.2(2)XB1	This command was implemented on the Cisco AS5850 platform.
12.2(8)T	This command was implemented on Cisco IAD2420 series integrated access devices (IADs). This command is not supported on the access servers in this release.
12.2(11)T	This command was implemented on Cisco AS5350, Cisco AS5400, and Cisco AS5850 platforms.

Examples

The following is output for the **debug backhaul-session-manager session all** command:

```
Router# debug backhaul-session-manager session all

Router# debug_bsm_command:DEBUG_BSM_SESSION_ALL

23:49:14:SESSION:XPORT:sig rcvd. session = 34, connid = 0x80BA12FC, sig = 5 (CONN-RESET)

23:49:14:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:CLOSE
23:49:14:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:14:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:OPEN_WAIT
23:49:14:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:19:SESSION:XPORT:sig rcvd. session = 34, connid = 0x80BA12FC, sig = 5 (CONN-RESET)

23:49:19:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:CLOSE
23:49:19:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:19:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:OPEN_WAIT
23:49:19:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:24:SESSION:XPORT:sig rcvd. session = 34, connid = 0x80BA12FC, sig = 5 (CONN-RESET)

23:49:24:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:CLOSE
23:49:24:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:24:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:OPEN_WAIT
23:49:24:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:29:SESSION:XPORT:sig rcvd. session = 34, connid = 0x80BA12FC, sig = 5 (CONN-RESET)

23:49:29:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:CLOSE
23:49:29:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:29:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:OPEN_WAIT
23:49:29:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:34:SESSION:XPORT:sig rcvd. session = 34, connid = 0x80BA12FC, sig = 5 (CONN-RESET)

23:49:34:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:CLOSE
23:49:34:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:49:34:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:OPEN_WAIT
```

```

23:49:34:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS
23:49:34:SESSION:XPORT:sig rcvd. session = 33, connid = 0x80BA14EC, sig = 1 (CONN-FAILED)
23:49:34:SESSION:STATE:(33) old-state:OPEN, new-state:CLOSE_WAIT

```

The following example displays output for the **debug backhaul-session-manager session state all** command:

```

Router# debug backhaul-session-manager session state all

Router# debug_bsm_command:DEBUG_BSM_SESSION_STATE_ALL

23:50:54:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:CLOSE
23:50:54:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

23:50:54:SESSION:STATE:(34) old-state:OPEN_WAIT, new-state:OPEN_WAIT
23:50:54:SESSION:STATE:(34) state:OPEN_WAIT, use-state:OOS

```

The following example displays output for the **debug backhaul-session-manager session xport all** command:

```

Router# debug backhaul-session-manager session xport all

Router# debug_bsm_command:DEBUG_BSM_SESSION_XPORT

23:51:39:SESSION:XPORT:sig rcvd. session = 34, connid = 0x80BA12FC, sig = 5 (CONN-RESET)
23:51:42:SESSION:XPORT:sig rcvd. session = 33, connid = 0x80BA14EC, sig = 5 (CONN-RESET)
23:51:44:SESSION:XPORT:sig rcvd. session = 34, connid = 0x80BA12FC, sig = 5 (CONN-RESET)

```

Related Commands

Command	Description
debug backhaul-session-manager set	Traces state changes and receives messages and events for all available session-sets or a specified session-set.

debug backhaul-session-manager set

To trace state changes and receive messages and events for all the available session sets or a specified session set, use the **debug backhaul-session-manager set** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug backhaul-session-manager set {all | name set-name}
```

```
no debug backhaul-session-manager set {all | name set-name}
```

Syntax Description

all	All available session sets.
name <i>set-name</i>	A specified session set.

Defaults

Debugging for backhaul session sets is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.2(2)T	Support for this command was introduced on the Cisco 7200 series routers.
12.2(4)T	This command was implemented on the Cisco 2600 series, Cisco 3600 series, and Cisco MC3810.
12.2(2)XB	This command was implemented on the Cisco AS5350 and Cisco AS5400.
12.2(2)XB1	This command was implemented on the Cisco AS5850 platform.
12.2(8)T	This command was implemented on Cisco IAD2420 series integrated access devices (IADs). This command is not supported on the access servers in this release.
12.2(11)T	This command was implemented on Cisco AS5350, Cisco AS5400, and Cisco AS5850 platforms.

Examples

The following is output for the **debug backhaul-session-manager set** command for all available session sets:

```
Router# debug backhaul-session-manager set all
```

```
Router# debug_bsm_command:DEBUG_BSM_SET_ALL
```

```
Function set_proc_event() is called
Session-Set :test-set
```

debug backhaul-session-manager set

```

Old State   :BSM_SET_OOS
New State   :BSM_SET_OOS
  Active-Grp :NONE
  Session-Grp :g-11
  Old State   :Group-None
  New State   :Group-None
  Event rcvd  :EVT_GRP_INS

BSM:Event BSM_SET_UP is sent to user
Session-Set :test-set
Old State   :BSM_SET_OOS
New State   :BSM_SET_ACTIVE_IS
  Active-Grp :g-11
  Session-Grp :g-11
  Old State   :Group-None
  New State   :Group-Active
  Event rcvd  :BSM_ACTIVE_TYPE

```

The following is output for the **debug backhaul-session-manager set name** *test-set* command:

```

Router# debug backhaul-session-manager set name set1

Router# debug_bsm_command:DEBUG_BSM_SET_NAME

Router# Function set_proc_event() is called
Session-Set :test-set
Old State   :BSM_SET_OOS
New State   :BSM_SET_OOS
  Active-Grp :NONE
  Session-Grp :g-11
  Old State   :Group-None
  New State   :Group-None
  Event rcvd  :EVT_GRP_INS

Router#BSM:Event BSM_SET_UP is sent to user
Session-Set :test-set
Old State   :BSM_SET_OOS
New State   :BSM_SET_ACTIVE_IS
  Active-Grp :g-11
  Session-Grp :g-11
  Old State   :Group-None
  New State   :Group-Active
  Event rcvd  :BSM_ACTIVE_TYPE

```

Related Commands

Command	Description
debug backhaul-session-manager session	Debugs all available sessions or a specified session.

debug backup

To monitor the transitions of an interface going down then back up, use the **debug backup** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug backup

no debug backup

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced.

Usage Guidelines

The **debug backup** command is useful for monitoring dual X.25 interfaces configured as primary and backup in a Telco data communication network (DCN).

Examples

The following example shows how to start the **debug backup** command:

```
Router# debug backup
```

Related Commands

Command	Description
backup active interface	Activates primary and backup lines on specific X.25 interfaces.
show backup	Displays interface backup status.

debug bert

To display information on the bit error rate testing (BERT) feature, use the **debug bert** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bert

no debug bert

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.0(2)XD	This command was introduced.

Usage Guidelines The **debug bert** command output is used primarily by Cisco technical support representatives. The **debug bert** command displays debugging messages for specific areas of executed code.

Examples The following is output from the **debug bert** command:

```
Router# debug bert

Bit Error Rate Testing debugging is on

Router# no debug bert

Bit Error Rate Testing debugging is off
```

Related Commands	Command	Description
	bert abort	Aborts a bit error rate testing session.
	bert controller	Starts a bit error rate test for a particular port on a Cisco AS5300 router.
	bert profile	Sets up various bit error rate testing profiles.

debug bgp ipv6 dampening

To display debugging messages for IPv6 Border Gateway Protocol (BGP) dampening, use the **debug bgp ipv6 dampening** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug bgp ipv6 dampening [prefix-list prefix-list-name]
```

```
no debug bgp ipv6 dampening [prefix-list prefix-list-name]
```

Syntax Description

prefix-list *prefix-list-name* (Optional) Name of an IPv6 prefix list.

Defaults

Debugging for IPv6 BGP dampening packets is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(13)T	The prefix-list keyword was added.

Usage Guidelines

The **debug bgp ipv6 dampening** command is similar to the **debug ip bgp dampening** command, except that it is IPv6-specific.

Use the **prefix-list** keyword and an argument to filter BGP IPv6 dampening debug information through an IPv6 prefix list.



Note

By default, the network server sends the output from **debug** commands and system error messages to the console. To redirect debugging output, use the **logging** command options within global configuration mode. Destinations are the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server.

Examples

The following is sample output from the **debug bgp ipv6 dampening** command:

```
Router# debug bgp ipv6 dampening

00:13:28:BGP(1):charge penalty for 2000:0:0:1::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:13:28:BGP(1):charge penalty for 2000:0:0:1:1::/80 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:13:28:BGP(1):charge penalty for 2000:0:0:5::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:16:03:BGP(1):charge penalty for 2000:0:0:1::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:16:03:BGP(1):flapped 2 times since 00:02:35. New penalty is 1892

00:18:28:BGP(1):suppress 2000:0:0:1:1::/80 path 2 1 for 00:27:30 (penalty 2671)
00:18:28:halflife-time 15, reuse/suppress 750/2000
00:18:28:BGP(1):suppress 2000:0:0:1:1::/64 path 2 1 for 00:27:20 (penalty 2664)
00:18:28:halflife-time 15, reuse/suppress 750/2000
```

The following example shows output for the **debug bgp ipv6 dampening** command filtered through the prefix list named “marketing”:

```
Router# debug bgp ipv6 dampening prefix-list marketing

00:16:08:BGP(1):charge penalty for 1234::/64 path 30 with halflife-time 15
reuse/suppress 750/2000
00:16:08:BGP(1):flapped 1 times since 00:00:00. New penalty is 10
```

[Table 24](#) describes the significant fields shown in the display.

Table 24 *debug bgp ipv6 dampening Field Descriptions*

Field	Description
penalty	Numerical value of 1000 assigned to a route by a router configured for route dampening in another autonomous system each time a route flaps. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. If the penalty exceeds the suppress limit, the route state changes from history to damp.
flapped	Number of times a route is available, then unavailable, or vice versa.
halflife-time	Amount of time (in minutes) by which the penalty is decreased after the route is assigned a penalty. The halflife-time value is half of the half-life period (which is 15 minutes by default). Penalty reduction happens every 5 seconds.
reuse	The limit by which a route is unsuppressed. If the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. Routes are unsuppressed at 10-second increments. Every 10 seconds, the router determines which routes are now unsuppressed and advertises them to the world.
suppress	Limit by which a route is suppressed. If the penalty exceeds this limit, the route is suppressed. The default value is 2000.

Table 24 *debug bgp ipv6 dampening Field Descriptions (continued)*

Field	Description
maximum suppress limit (not shown in sample output)	Maximum amount of time (in minutes) a route is suppressed. The default value is four times the half-life period.
damp state (not shown in sample output)	State in which the route has flapped so often that the router will not advertise this route to BGP neighbors.

Related Commands

Command	Description
debug bgp ipv6 updates	Displays debugging messages for IPv6 BGP update packets.

debug bgp ipv6 updates

To display debugging messages for IPv6 Border Gateway Protocol (BGP) update packets, use the **debug bgp ipv6 updates** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bgp ipv6 updates [*ipv6-address*] [**prefix-list** *prefix-list-name*] [**in** | **out**]

no debug bgp ipv6 updates [*ipv6-address*] [**prefix-list** *prefix-list-name*] [**in** | **out**]

Syntax Description

<i>ipv6-address</i>	(Optional) The IPv6 address of a BGP neighbor. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
prefix-list <i>prefix-list-name</i>	(Optional) Name of an IPv6 prefix list.
in	(Optional) Indicates inbound updates.
out	(Optional) Indicates outbound updates.

Defaults

Debugging for IPv6 BGP update packets is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(13)T	The prefix-list keyword was added.

Usage Guidelines

The **debug bgp ipv6 updates** command is similar to the **debug ip bgp updates** command, except that it is IPv6-specific.

Use the **prefix-list** keyword to filter BGP IPv6 updates debugging information through an IPv6 prefix list.



Note

By default, the network server sends the output from **debug** commands and system error messages to the console. To redirect debugging output, use the **logging** command options within global configuration mode. Destinations are the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server.

Examples

The following is sample output from the **debug bgp ipv6 updates** command:

```
Router# debug bgp ipv6 updates

14:04:17:BGP(1):2000:0:0:2::2 computing updates, afi 1, neighbor version 0, table version
1, starting at ::
14:04:17:BGP(1):2000:0:0:2::2 update run completed, afi 1, ran for 0ms, neighbor version
0, start version 1, throttled to 1
14:04:19:BGP(1):sourced route for 2000:0:0:2::1/64 path #0 changed (weight 32768)
14:04:19:BGP(1):2000:0:0:2::1/64 route sourced locally
14:04:19:BGP(1):2000:0:0:2:1::/80 route sourced locally
14:04:19:BGP(1):2000:0:0:3::2/64 route sourced locally
14:04:19:BGP(1):2000:0:0:4::2/64 route sourced locally
14:04:22:BGP(1):2000:0:0:2::2 computing updates, afi 1, neighbor version 1, table version
6, starting at ::
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (format) 2000:0:0:2::1/64, next 2000:0:0:2::1,
metric 0, path
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (format) 2000:0:0:2:1::/80, next 2000:0:0:2::1,
metric 0, path
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (prepend, chgflags:0x208) 2000:0:0:3::2/64, next
2000:0:0:2::1, metric 0, path
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (prepend, chgflags:0x208) 2000:0:0:4::2/64, next
2000:0:0:2::1, metric 0, path
```

The following is sample output from the **debug bgp ipv6 updates** command filtered through the prefix list named "sales":

```
Router# debug bgp ipv6 updates prefix-list sales

00:18:26:BGP(1):2000:8493:1::2 send UPDATE (prepend, chgflags:0x208) 7878:7878::/64, next
2F02:3000::36C, metric 0, path
```

[Table 25](#) describes the significant fields shown in the display.

Table 25 *debug bgp ipv6 updates Field Descriptions*

Field	Description
BGP(1):	BGP debugging for address family index (afi) 1.
afi	Address family index.
neighbor version	Version of the BGP table on the neighbor from which the update was received.
table version	Version of the BGP table on the router from which you entered the debug bgp ipv6 updates command.
starting at	Starting at the network layer reachability information (NLRI). BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address, community values, and other information.
route sourced locally	Indicates that a route is sourced locally and that updates are not sent for the route.
send UPDATE (format)	Indicates that an update message for a reachable network should be formatted. Addresses include prefix and next hop.
send UPDATE (prepend, chgflags:0x208)	Indicates that an update message about a path to a BGP peer should be written.

■ debug bgp ipv6 updates**Related Commands**

Command	Description
debug bgp ipv6 dampening	Displays debugging messages for IPv6 BGP dampening packets.

debug bgp nsap

To enable the display of Border Gateway Protocol (BGP) debugging information specific to the network service access point (NSAP) address family, use the **debug bgp nsap** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bgp nsap

no debug bgp nsap

Syntax Description

This command has no arguments or keywords.

Defaults

Debugging of BGP NSAP address-family code is not enabled.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

The **debug bgp nsap** command is similar to the **debug ip bgp** command, except that it is specific to the NSAP address family.



Note

By default, the network server sends the output from debug commands and system error messages to the console. To redirect debug output, use the **logging** command options within global configuration mode. Destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server.

Examples

The following example shows output for the **debug bgp nsap** command. The BGP(4) identifies that BGP version 4 is operational.

```
Router# debug bgp nsap

00:46:46: BGP(4): removing CLNS route to 49.0101
00:46:46: BGP(4): removing CLNS route to 49.0303
00:46:46: BGP(4): removing CLNS route to 49.0404
00:46:46: BGP(4): 10.1.2.1 removing CLNS route 49.0101.1111.1111.1111.1111.00 to
eBGP-neighbor
00:46:46: BGP(4): 10.2.4.4 removing CLNS route 49.0303.4444.4444.4444.4444.00 to
eBGP-neighbor
00:46:59: BGP(4): Applying map to find origin for prefix 49.0202.2222
00:46:59: BGP(4): Applying map to find origin for prefix 49.0202.3333
```

Related Commands

Command	Description
debug bgp nsap dampening	Displays debug messages for BGP NSAP prefix dampening events.
debug bgp nsap updates	Displays debug messages for BGP NSAP prefix update packets.

debug bgp nsap dampening

To display debug messages for Border Gateway Protocol (BGP) network service access point (NSAP) prefix address dampening, use the **debug bgp nsap dampening** command in privileged EXEC mode. To disable debug messages for NSAP BGP dampening, use the **no** form of this command.

```
debug bgp nsap dampening [filter-list access-list-number]
```

```
no debug bgp nsap dampening [filter-list access-list-number]
```

Syntax Description

filter-list access-list-number (Optional) Displays debug messages for BGP NSAP dampening events that match the access list. The acceptable access list number range is from 1 to 199.

Defaults

Debugging for BGP NSAP dampening events is not enabled.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

The **debug bgp nsap dampening** command is similar to the **debug ip bgp dampening** command, except that it is specific to the NSAP address family.



Note

By default, the network server sends the output from **debug** commands and system error messages to the console. To redirect debug output, use the **logging** command options within global configuration mode. Destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server.

Examples

The following example shows output for the **debug bgp nsap dampening** command:

```
Router# debug bgp nsap dampening
```

```
16:21:34: BGP(4): Dampening route-map modified.
```

Only one line of output is displayed unless the **bgp dampening** command is configured with a route map in NSAP address family configuration mode. The following example shows output for the **debug bgp nsap dampening** command when a route map is configured:

```
20:07:19: BGP(4): charge penalty for 49.0404 path 65202 65404 with halflife-time 15
reuse/suppress 750/2000
```

```
20:07:19: BGP(4): flapped 1 times since 00:00:00. New penalty is 1000
```

```
20:08:59: BGP(4): charge penalty for 49.0404 path 65202 65404 with halflife-time 15
reuse/suppress 750/2000
```

```

20:08:59: BGP(4): flapped 2 times since 00:01:39. New penalty is 1928

20:10:04: BGP(4): charge penalty for 49.0404 path 65202 65404 with halflife-time 15
reuse/suppress 750/2000
20:10:04: BGP(4): flapped 3 times since 00:02:44. New penalty is 2839

20:10:48: BGP(4): suppress 49.0404 path 65202 65404 for 00:28:10 (penalty 2752)
20:10:48: halflife-time 15, reuse/suppress 750/2000

```

Table 26 describes the significant fields shown in the display.

Table 26 *debug bgp nsap dampening Field Descriptions*

Field	Description
penalty	Numerical value of 1000 assigned to a route by a router configured for route dampening in another autonomous system each time a route flaps. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. If the penalty exceeds the suppress limit, the route state changes from history to damp.
halflife-time	Amount by which the penalty is decreased after the route is assigned a penalty. The half-life-time value is half of the half-life period (which is 15 minutes by default). Penalty reduction occurs every 5 seconds.
flapped	Number of times a route is available, then unavailable, or vice versa.
reuse	The limit by which a route is unsuppressed. If the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. Unsuppressing of routes occurs at 10-second increments. Every 10 seconds, the router learns which routes are now unsuppressed and advertises them throughout the network.
suppress	Limit by which a route is suppressed. If the penalty exceeds this limit, the route is suppressed. The default value is 2000.
maximum suppress limit (not shown in sample output)	Maximum amount of time a route is suppressed. The default value is four times the half-life period.
damp state (not shown in sample output)	State in which the route has flapped so often that the router will not advertise this route to BGP neighbors.

Related Commands

Command	Description
debug bgp nsap	Displays debug messages for BGP NSAP packets.
debug bgp nsap updates	Displays debug messages for BGP NSAP update events.

debug bgp nsap updates

To display debug messages for Border Gateway Protocol (BGP) network service access point (NSAP) prefix address update packets, use the **debug bgp nsap updates** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug bgp nsap updates [ip-address] [in | out] [filter-set clns-filter-set-name]
```

```
no debug bgp nsap updates [ip-address] [in | out] [filter-set clns-filter-set-name]
```

Syntax Description

<i>ip-address</i>	(Optional) The IP address of a BGP neighbor.
in	(Optional) Indicates inbound updates.
out	(Optional) Indicates outbound updates.
filter-set <i>clns-filter-set-name</i>	(Optional) Name of a Connectionless Network Service (CLNS) filter set.

Defaults

Debugging for BGP NSAP prefix update packets is not enabled.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

The **debug bgp nsap updates** command is similar to the **debug ip bgp updates** command, except that it is specific to the NSAP address family.

Use the *ip-address* argument to display the BGP update debug messages for a specific BGP neighbor.

Use the *clns-filter-set-name* argument to display the BGP update debug messages for a specific NSAP prefix.



Note

By default, the network server sends the output from **debug** commands and system error messages to the console. To redirect debug output, use the **logging** command options within global configuration mode. Destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server.

Examples

The following example shows output for the **debug bgp nsap updates** command:

```
Router# debug bgp nsap updates

02:13:45: BGP(4): 10.0.3.4 send UPDATE (format) 49.0101, next
49.0303.3333.3333.3333.3333.00, metric 0, path 65202 65101
02:13:45: BGP(4): 10.0.3.4 send UPDATE (format) 49.0202, next
49.0303.3333.3333.3333.3333.00, metric 0, path 65202
02:13:45: BGP(4): 10.0.3.4 send UPDATE (format) 49.0303, next
49.0303.3333.3333.3333.3333.00, metric 0, path
02:13:45: BGP(4): 10.0.2.2 send UPDATE (format) 49.0404, next
49.0303.3333.3333.3333.3333.00, metric 0, path 65404
```

[Table 27](#) describes the significant fields shown in the display.

Table 27 *debug bgp nsap updates Field Descriptions*

Field	Description
BGP(4):	BGP debug for address family index (afi) 4.
route sourced locally (not shown in display)	Indicates that a route is sourced locally and that updates are not sent for the route.
send UPDATE (format)	Indicates that an update message for a reachable network should be formatted. Addresses include NSAP prefix and next hop.
rcv UPDATE (not shown in display)	Indicates that an update message about a path to a BGP peer has been received. Addresses include NSAP prefix.

Related Commands

Command	Description
debug bgp nsap	Displays debug messages for BGP NSAP packets.
debug bgp nsap dampening	Displays debug messages for BGP NSAP prefix dampening events.

debug bri-interface

To display debugging information on ISDN BRI routing activity, use the **debug bri-interface** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bri-interface

no debug bri-interface

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The **debug bri-interface** command indicates whether the ISDN code is enabling and disabling the B channels when attempting an outgoing call. This command is available for the low-end router products that have a multi-BRI network interface module installed.



Caution

Because the **debug bri-interface** command generates a substantial amount of output, use it only when traffic on the IP network is low, so other activity on the system is not adversely affected.

Examples

The following is sample output from the **debug bri-interface** command:

```
Router# debug bri-interface
```

```
BRI: write_sid: wrote 1B for subunit 0, slot 1.
BRI: write_sid: wrote 15 for subunit 0, slot 1.
BRI: write_sid: wrote 17 for subunit 0, slot 1.
BRI: write_sid: wrote 6 for subunit 0, slot 1.
BRI: write_sid: wrote 8 for subunit 0, slot 1.
BRI: write_sid: wrote 11 for subunit 0, slot 1.
BRI: write_sid: wrote 13 for subunit 0, slot 1.
BRI: write_sid: wrote 29 for subunit 0, slot 1.
BRI: write_sid: wrote 1B for subunit 0, slot 1.
BRI: write_sid: wrote 15 for subunit 0, slot 1.
BRI: write_sid: wrote 17 for subunit 0, slot 1.
BRI: write_sid: wrote 20 for subunit 0, slot 1.
BRI: Starting Power Up timer for unit = 0.
BRI: write_sid: wrote 3 for subunit 0, slot 1.
BRI: Starting T3 timer after expiry of PUP timeout for unit = 0, current state is F4.
BRI: write_sid: wrote FF for subunit 0, slot 1.
BRI: Activation for unit = 0, current state is F7.
BRI: enable channel B1
BRI: write_sid: wrote 14 for subunit 0, slot 1.

%LINK-3-UPDOWN: Interface BRI0: B-Channel 1, changed state to up
%LINK-5-CHANGED: Interface BRI0: B-Channel 1, changed state to up.!!!
BRI: disable channel B1
BRI: write_sid: wrote 15 for subunit 0, slot 1.
```

```
%LINK-3-UPDOWN: Interface BRI0: B-Channel 1, changed state to down
%LINK-5-CHANGED: Interface BRI0: B-Channel 1, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0: B-Channel 1, changed state to down
```

The following line indicates that an internal command was written to the interface controller. The subunit identifies the first interface in the slot.

```
BRI: write_sid: wrote 1B for subunit 0, slot 1.
```

The following line indicates that the power-up timer was started for the named unit:

```
BRI: Starting Power Up timer for unit = 0.
```

The following lines indicate that the channel or the protocol on the interface changed state:

```
%LINK-3-UPDOWN: Interface BRI0: B-Channel 1, changed state to up
%LINK-5-CHANGED: Interface BRI0: B-Channel 1, changed state to up.!!!
%LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0: B-Channel 1, changed state to down
```

The following line indicates that the channel was disabled:

```
BRI: disable channel B1
```

Lines of output not described are for use by support staff only.

Related Commands

Command	Description
debug isdn event	Displays ISDN events occurring on the user side (on the router) of the ISDN interface.
debug isdn q921	Displays data link-layer (Layer 2) access procedures that are taking place at the router on the D channel (LSPD).
debug isdn q931	Displays information about call setup and teardown of ISDN network connections (Layer 3) between the local router (user side) and the network.

debug bsc event

To display all events occurring in the Binary Synchronous Communications (Bisync) feature, use the **debug bsc event** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bsc event *[number]*

no debug bsc event *[number]*

Syntax Description

number (Optional) Group number.

Command Modes

Privileged EXEC

Usage Guidelines

This command traces all interfaces configured with a **bsc protocol-group** *number* command.

Examples

The following is sample output from the **debug bsc event** command:

```
Router# debug bsc event

BSC: Serial2          POLLEE-FSM inp:E_LineFail old_st:CU_Down new_st:TCU_EOFfile
BSC: Serial2          POLLEE-FSM inp:E_LineFail old_st:CU_Down new_st:TCU_EOFfile
BSC: Serial2          POLLEE-FSM inp:E_LineFail old_st:CU_Down new_st:TCU_EOFfile
0:04:32: BSC: Serial2 :SDI-rx: 9 bytes
BSC: Serial2          POLLEE-FSM inp:E_RxEtx old_st:CU_Down new_st:TCU_EOFfile
0:04:32: BSC: Serial2 :SDI-rx: 5 bytes
BSC: Serial2          POLLEE-FSM inp:E_RxEnq old_st:CU_Down new_st:TCU_EOFfile
BSC: Serial2          POLLEE-FSM inp:E_Timeout old_st:CU_Down new_st:TCU_InFile
BSC: Serial2          POLLEE-FSM inp:E_Timeout old_st:CU_Idle new_st:TCU_InFile
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2, changed state to up
%LINK-3-UPDOWN: Interface Serial2, changed state to up
BSC: Serial2          POLLEE-FSM inp:E_Timeout old_st:CU_Idle new_st:TCU_InFile
0:04:35: BSC: Serial2 :SDI-rx: 9 bytes
BSC: Serial2          POLLEE-FSM inp:E_RxEtx old_st:CU_Idle new_st:TCU_InFile
0:04:35: BSC: Serial2 :SDI-rx: 5 bytes
BSC: Serial2          POLLEE-FSM inp:E_RxEnq old_st:CU_Idle new_st:TCU_InFile
0:04:35: BSC: Serial2 :NDI-rx: 3 bytes
```

Related Commands

Command	Description
debug bsc packet	Displays all frames traveling through the Bisync feature.
debug bstun events	Displays BSTUN connection events and status.

debug bsc packet

To display all frames traveling through the Binary Synchronous Communications (Bisync) feature, use the **debug bsc packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bsc packet [*group number*] [*buffer-size bytes*]

no debug bsc packet [*group number*] [*buffer-size bytes*]

Syntax Description

group number	(Optional) Group number.
buffer-size bytes	(Optional) Number of bytes displayed per packet (defaults to 20).

Defaults

The default number of bytes displayed is 20.

Command Modes

Privileged EXEC

Usage Guidelines

This command traces all interfaces configured with a **bsc protocol-group number** command.

Examples

The following is sample output from the **debug bsc packet** command:

```
Router# debug bsc packet

0:23:33: BSC: Serial2      :NDI-rx : 27 bytes 401A400227F5C31140C11D60C8C5D3D3D51D4013
0:23:33: BSC: Serial2      :SDI-tx  : 12 bytes 00323237FF3232606040402D
0:23:33: BSC: Serial2      :SDI-rx  : 2 bytes 1070
0:23:33: BSC: Serial2      :SDI-tx  : 27 bytes 401A400227F5C31140C11D60C8C5D3D3D51D4013
0:23:33: BSC: Serial2      :SDI-rx  : 2 bytes 1061
0:23:33: BSC: Serial2      :SDI-tx  : 5 bytes 00323237FF
```

Related Commands

Command	Description
debug bsc event	Displays all events occurring in the Bisync feature.
debug bstun events	Displays BSTUN connection events and status.

debug bstun events

To display BSTUN connection events and status, use the **debug bstun events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bstun events [*number*]

no debug bstun events [*number*]

Syntax Description

number (Optional) Group number.

Command Modes

Privileged EXEC

Usage Guidelines

When you enable the **debug bstun events** command, messages showing connection establishment and other overall status messages are displayed.

You can use the **debug bstun events** command to assist you in determining whether the BSTUN peers are configured correctly and are communicating. For example, if you enable the **debug bstun packet** command and you do not see any packets, you may want to enable event debugging.



Note

Also refer to the **debug bsc packet** and **debug bsc event** commands. Currently, these two commands support the only protocol working through the BSTUN tunnel. Sometimes frames do not go through the tunnel because they have been discarded at the Bisync protocol level.

Examples

The following is sample output from the **debug bstun events** command of keepalive messages working correctly. If the routers are configured correctly, at least one router will show reply messages.

```
Router# debug bstun events
```

```
BSTUN: Received Version Reply opcode from (all[2])_172.16.12.2/1976 at 1360
BSTUN: Received Version Request opcode from (all[2])_172.16.12.2/1976 at 1379
BSTUN: Received Version Reply opcode from (all[2])_172.16.12.2/1976 at 1390
```



Note

In a scenario where there is constantly loaded bidirectional traffic, you might not see keepalive messages because they are sent only when the remote end has been silent for the keepalive period.

The following is sample output from the **debug bstun events** output of an event trace in which the wrong TCP address has been specified for the remote peer. These are non-keepalive related messages.

```
Router# debug bstun events
```

```
BSTUN: Change state for peer (C1[1])172.16.12.22/1976 (closed->opening)
BSTUN: Change state for peer (C1[1])172.16.12.22/1976 (opening->open wait)
%BSTUN-6-OPENING: CONN: opening peer (C1[1])172.16.12.22/1976, 3
BSTUN: tcpd sender in wrong state, dropping packet
BSTUN: tcpd sender in wrong state, dropping packet
BSTUN: tcpd sender in wrong state, dropping packet
```

Related Commands

Command	Description
debug bsc event	Displays all events occurring in the Bisync feature.
debug bsc packet	Displays all frames traveling through the Bisync feature.
debug bstun packet	Displays packet information on packets traveling through the BSTUN links.

debug bstun packet

To display packet information on packets traveling through the BSTUN links, use the **debug bstun packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug bstun packet [group number] [buffer-size bytes]
```

```
no debug bstun packet [group number] [buffer-size bytes]
```

Syntax Description

group number	(Optional) BSTUN group number.
buffer-size bytes	(Optional) Number of bytes displayed per packet (defaults to 20).

Defaults

The default number of bytes displayed is 20.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debug bstun packet** command:

```
Router# debug bstun packet
```

```
BSTUN bsc-local-ack: 0:00:00 Serial2          SDI: Addr: 40 Data: 02C1C1C1C1C1C1C1C1C1
BSTUN bsc-local-ack: 0:00:00 Serial2          SDI: Addr: 40 Data: 02C1C1C1C1C1C1C1C1C1
BSTUN bsc-local-ack: 0:00:06 Serial2          NDI: Addr: 40 Data: 0227F5C31140C11D60C8
```

Related Commands

Command	Description
debug bstun events	Displays BSTUN connection events and status.

debug bundle errors

To enable the display of information on bundle errors, use the **debug bundle errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bundle errors

no debug bundle errors

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.0(3)T	This command was introduced.

Usage Guidelines Use this command to enable the display of error information for a bundle, such as reports of inconsistent mapping in the bundle.

Related Commands	Command	Description
	bump	Configures the bumping rules for a VC class that can be assigned to a VC bundle.
	bundle	Creates a bundle or modifies an existing bundle to enter bundle configuration mode.
	debug bundle events	Enables display of bundle events when use occurs.

debug bundle events

To enable display of bundle events when use occurs, use the **debug bundle events** command in privileged EXEC mode. To disable the display, use the **no** form of this command.

debug bundle events

no debug bundle events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)T	This command was introduced.

Usage Guidelines

Use this command to enable the display of bundle events, such as occurrences of VC bumping, when bundles were brought up, when they were taken down, and so forth.

Related Commands

Command	Description
debug bstun packet	Enables the display of information on bundle errors.