debug backhaul-session-manager session through debug channel packets

debug backhaul-session-manager session through debug channel packets

debug backhaul-session-manager session

To debug all the available sessions or a specified session, use the **debugbackhaul-session-managersession** 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*}

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.
session-id	A specified session.

Command Default

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

Examples

The following is output for the **debugbackhaul-session-managersessionall**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 **debugbackhaul-session-managersessionstateall** 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 **debugbackhaul-session-managersessionxportall** 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 Comma

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.

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 **debugbackhaul-session-managerset** 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 *set-name*	A specified session set.

Command Default

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

Examples

The following is output for the **debugbackhaul-session-managerset**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
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 **debugbackhaul-session-managersetnameset1**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 **debugbackup**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.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

The **debugbackup**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 **debugbackup** 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 **debugbert** 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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

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

Examples

The following is output from the **debugbert** 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 bfd

To display debugging messages about Bidirectional Forwarding Detection (BFD), use the **debug bfd** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

Cisco IOS Release 12.2(18)SXE, 12.4(4)T, and 12.2(33)SRA

debug bfd {**event**| **packet** [*ip-address*| *ipv6-address*]}

no debug bfd {**event**| **packet** [*ip-address*| *ipv6-address*]}

Cisco IOS Release 12.0(31)S

debug bfd {**event**| **packet** [*ip-address*]| **ipc-error**| **ipc-event**| **oir-error**| **oir-event**}

no debug bfd {**event**| **packet** [*ip-address*]| **ipc-error**| **ipc-event**| **oir-error**| **oir-event**}

Syntax Description

event	Displays debugging information about BFD state transitions.
packet	Displays debugging information about BFD control packets.
ip-address	(Optional) Displays debugging information about BFD only for the specified IP address.
ipv6-address	(Optional) Displays debugging information about BFD only for the specified IPv6 address.
ipc-error	(Optional) Displays debugging information with interprocess communication (IPC) errors on the Route Processor (RP) and line card (LC).
ipc-event	(Optional) Displays debugging information with IPC events on the RP and LC.
oir-error	(Optional) Displays debugging information with online insertion and removal (OIR) errors on the RP and LC.
oir-event	(Optional) Displays debugging information with OIR events on the RP and LC.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(18)SXE	This command was introduced.
12.0(31)S	This command was integrated into Cisco IOS Release 12.0(31)S.
12.4(4)T	This command was integrated into Cisco IOS Release 12.4(4)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SRE	This command was modified. Support for IPv6 was added.
15.1(2)T	This command was modified. Support for IPv6 was added to Cisco IOS Release 15.1(2)T.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.
15.1(1)SY	This command was modified. Support for IPv6 was added to Cisco IOS Release 15.1(1)SY.

Usage Guidelines

The **debug bfd** command can be used to troubleshoot the BFD feature.

Note Because BFD is designed to send and receive packets at a very high rate of speed, consider the potential effect on system resources before enabling this command, especially if there are a large number of BFD peers. The **debug bfd packet** command should be enabled only on a live network at the direction of Cisco Technical Assistance Center personnel.

Examples

The following example shows output from the **debug bfd packet** command. The IP address has been specified in order to limit the packet information to one interface:

```
Router# debug bfd packet 172.16.10.5
BFD packet debugging is on
*Jan 26 14:47:37.645: Tx*IP: dst 172.16.10.1, plen 24. BFD: diag 2, St/D/P/F (1/0/0/0),
mult 5, len 24, loc/rem discr 1 1, tx 1000000, rx 1000000 100000, timer 1000 ms, #103
*Jan 26 14:47:37.645: %OSPF-5-ADJCHG: Process 10, Nbr 172.16.10.12 on Ethernet1/4 from FULL
 to DOWN, Neighbor Down: BFD node down
*Jan 26 14:47:50.685: %OSPF-5-ADJCHG: Process 10, Nbr 172.16.10.12 on Ethernet1/4 from
LOADING to FULL, Loading Done
*Jan 26 14:48:00.905: Rx  IP: src 172.16.10.1, plen 24. BFD: diag 0, St/D/P/F (1/0/0/0),
mult 4, len 24, loc/rem discr 2 1, tx 1000000, rx 1000000 100000, timer 4000 ms, #50
*Jan 26 14:48:00.905: Tx IP: dst 172.16.10.1, plen 24. BFD: diag 2, St/D/P/F (2/0/0/0),
mult 5, len 24, loc/rem discr 1 2, tx 1000000, rx 1000000 100000, timer 1000 ms, #131
*Jan 26 14:48:00.905: Rx  IP: src 172.16.10.1, plen 24. BFD: diag 0, St/D/P/F (3/0/0/0),
mult 4, len 24, loc/rem discr 2 1, tx 1000000, rx 1000000 100000, timer 4000 ms, #51
*Jan 26 14:48:00.905: Tx IP: dst 172.16.10.1, plen 24. BFD: diag 0, St/D/P/F (3/0/0/0),
mult 5, len 24, loc/rem discr 1 2, tx 1000000, rx 1000000 100000, timer 1000 ms, #132
```

The following example shows output from the **debug bfd event** command when an interface between two BFD neighbor routers fails and then comes back online:

```
Router# debug bfd event
22:53:48: BFD: bfd_neighbor - action:DESTROY, proc:1024, idb:FastEthernet0/1,
neighbor:172.16.10.2
22:53:48: BFD: bfd_neighbor - action:DESTROY, proc:512, idb:FastEthernet0/1,
neighbor:172.16.10.2
22:53:49: Session [172.16.10.1,172.16.10.2,Fa0/1,1], event DETECT TIMER EXPIRED, state UP
-> FAILING
.
.
.
22:56:35: BFD: bfd_neighbor - action:CREATE, proc:1024, idb:FastEthernet0/1,
neighbor:172.16.10.2
22:56:37: Session [172.16.10.1,172.16.10.2,Fa0/1,1], event RX IHY 0, state FAILING -> DOWN
22:56:37: Session [172.16.10.1,172.16.10.2,Fa0/1,1], event RX IHY 0, state DOWN -> INIT
22:56:37: Session [172.16.10.1,172.16.10.2,Fa0/1,1], event RX IHY 1, state INIT -> UP
```

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

Table 1: debug bfd event Field Descriptions

Field	Description
bfd_neighbor - action:DESTROY	The BFD neighbor will tear down the BFD session.
Session [172.16.10.1, 172.16.10.2, Fa0/1,1]	IP addresses of the BFD neighbors holding this session that is carried over FastEthernet interface 0/1.
event DETECT TIMER EXPIRED	The BFD neighbor has not received BFD control packets within the negotiated interval and the detect timer has expired.
state UP -> FAILING	The BFD event state is changing from Up to Failing.
Session [172.16.10.1, 172.16.10.2, Fa0/1,1], event RX IHY 0	The BFD session between the neighbors indicated by the IP addresses that is carried over FastEthernet interface 0/1 is changing state from Failing to Down. The I Hear You (IHY) bit value is shown as 0 to indicate that the remote system is tearing down the BFD session.
event RX IHY 0, state DOWN -> INIT	The BFD session is still considered down, and the IHY bit value still is shown as 0, and the session state changes from DOWN to INIT to indicate that the BFD session is again initializing, as the interface comes back up.
event RX IHY 1, state INIT -> UP	The BFD session has been reestablished, and the IHY bit value changes to 1 to indicate that the session is live. The BFD session state changes from INIT to UP.

The following example shows output from the **debug bfd packet** command when an interface between two BFD neighbor routers fails and then comes back online. The diagnostic code changes from 0 (No Diagnostic) to 1 (Control Detection Time Expired) because no BFD control packets could be sent (and therefore detected by the BFD peer) after the interface fails. When the interface comes back online, the diagnostic code changes back to 0 to signify that BFD packets can be sent and received by the BFD peers.

```
Router# debug bfd packet
23:03:25: Rx  IP: src 172.16.10.2, plen 24. BFD: diag 0, H/D/P/F (0/0/0/0), mult 3, len 24,
 loc/rem discr 5 1, tx 1000000, rx 100007
23:03:25: Tx IP: dst 172.16.10.2, plen 24. BFD: diag 1, H/D/P/F (0/0/0/0), mult 5, len 24,
 loc/rem discr 1 5, tx 1000000, rx 1000008
23:03:25: Tx IP: dst 172.16.10.2, plen 24. BFD: diag 1, H/D/P/F (1/0/0/0), mult 5, len 24,
 loc/rem discr 1 5, tx 1000000, rx 1000009
```

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

Table 2: debug bfd packet Field Descriptions

Field	Description
Rx IP: src 172.16.10.2	The router has received this BFD packet from the BFD router with source address 172.16.10.2.
plen 24	Length of the BFD control packet, in bytes.
diag 0	A diagnostic code specifying the local system's reason for the last transition of the session from Up to some other state. State values are as follows: • 0--No Diagnostic • 1--Control Detection Time Expired • 2--Echo Function Failed • 3--Neighbor Signaled Session Down • 4--Forwarding Plane Reset • 5--Path Down • 6--Concentrated Path Down • 7--Administratively Down

Field	Description
H/D/P/F (0/0/0/0)	H bit--Hear You bit. This bit is set to 0 if the transmitting system either is not receiving BFD packets from the remote system or is tearing down the BFD session. During normal operation the I Hear You bit is set to 1. D bit--Demand Mode bit. If the Demand Mode bit set, the transmitting system wants to operate in demand mode. BFS has two modes--asynchronous and demand. The Cisco implementation of BFD supports only asynchronous mode. P bit--Poll bit. If the Poll bit is set, the transmitting system is requesting verification of connectivity or of a parameter change. F bit--Final bit. If the Final bit is set, the transmitting system is responding to a received BFC control packet that had a Poll (P) bit set.
mult 3	Detect time multiplier. The negotiated transmit interval, multiplied by the detect time multiplier, determines the detection time for the transmitting system in BFD asynchronous mode. The detect time multiplier is similar to the hello multiplier in IS-IS, which is used to determine the hold timer: (hellointerval) * (hellomultiplier) = hold timer. If a hello packet is not received within the hold-timer interval, a failure has occurred. Similarly, for BFD: (transmit interval) * (detect multiplier) = detect timer. If a BFD control packet is not received from the remote system within the detect-timer interval, a failure has occurred.
len 24	The BFD packet length.
loc/rem discr 5 1	The values for My Discriminator (local) and Your Discriminator (remote) BFD neighbors. • My Discriminator--Unique, nonzero discriminator value generated by the transmitting system, used to demultiplex multiple BFD sessions between the same pair of systems. • Your Discriminator--The discriminator received from the corresponding remote system. This field reflects the received value of My Discriminator, or is zero if that value is unknown.

Field	Description
tx 1000000	Desired minimum transmit interval.
rx 100007	Required minimum receive interval.

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 messages for IPv6 BGP dampening, use the **no** form of this command.

debug bgp ipv6 {**unicast**| **multicast**} **dampening** [**prefix-list** *prefix-list-name*]

no debug bgp ipv6 {**unicast**| **multicast**} **dampening** [**prefix-list** *prefix-list-name*]

Syntax Description

unicast	Specifies IPv6 unicast address prefixes.
multicast	Specifies IPv6 multicast address prefixes.
prefix-list *prefix-list-name*	(Optional) Name of an IPv6 prefix list.

Command Default

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.
12.0(24)S	The **prefix-list** keyword was added.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

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::/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 2001:0DB8::/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
```

The table below describes the fields shown in the display.

Table 3: 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.

Field	Description
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.
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 messages for IPv6 BGP update packets, use the **no** form of this command.

debug bgp ipv6 {**unicast**| **multicast**} **updates** [*ipv6-address*] [**prefix-list** *prefix-list-name*] [**in**| **out**]

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

Syntax Description

unicast	Specifies IPv6 unicast address prefixes.
multicast	Specifies IPv6 multicast address prefixes.
ipv6-address	(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 *prefix-list-name*	(Optional) Name of an IPv6 prefix list.
in	(Optional) Indicates inbound updates.
out	(Optional) Indicates outbound updates.

Command Default

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.
12.0(24)S	The **prefix-list** keyword was added.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Release	Modification
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

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. For complete information on **debug** commands and redirecting debugging output, refer to the Release 12.2 *Cisco IOS Debug Command Reference* .

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
2001:0DB8::36C, metric 0, path
```

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

Table 4: 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.

Related Commands

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

debug bgp l2vpn vpls updates

To enable debugging of the L2VPN VPLS address family updates from the BGP table, use the **debug bgp l2vpn vpls updates** command in privileged EXEC mode. To disable the display of the messages, use the **no** form of this command.

debug bgp l2vpn vpls updates [*access-list* | *expanded-access-list* | *bgp-neighbor-address* | **events** | {**in** | **out** }]

nodebug bgp l2vpn vpls updates [*access-list* | *expanded-access-list* | *bgp-neighbor-address* | **events** | {**in** | **out** }]

Syntax Description

access-list	(Optional) Number of an access list used to filter debugging messages. The range is from 1 to 199.
expanded-access-list	(Optional) Number of an expanded access list used to filter debugging messages. The range is from 1300 to 2699.
bgp-neighbor-address	(Optional) BGP neighbor address in the format A.B.C.D.
events	(Optional) Specifies debugging messages for BGP update events.
in	Specifies debugging messages for inbound BGP update information.
out	Specifies debugging messages for outbound BGP update information.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.8S	This command was introduced.

Examples

The following shows how to enable the **debug bgp l2vpn vpls updates** command:

```
Device> enable
Device# debug bgp l2vpn vpls updates
BGP updates debugging is on for address family: L2VPN Vpls
```

Related Commands

Command	Description
debug ip bgp updates	Displays information about the processing of BGP updates.
show bgp l2vpn vpls	Displays L2VPN VPLS address family information from the BGP table.

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 **debugbgpnsap** 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.

Command Default

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.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
Cisco IOS XE 2.6	This command was integrated into Cisco IOS XE Release 2.6.

Usage Guidelines

The **debugbgpnsap**command is similar to the **debugipbgp**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 **debugbgpnsap**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 **debugbgpnsapdampening** command in privileged EXEC mode. To disable debugging output, 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.

Command Default

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.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
Cisco IOS XE 2.6	This command was integrated into Cisco IOS XE Release 2.6.

Usage Guidelines

The **debugbgpnsapdampening**command is similar to the **debugipbgpdampening**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 **debugbgpnsapdampening**command:

```
Router# debug bgp nsap dampening
16:21:34: BGP(4): Dampening route-map modified.
```

Only one line of output is displayed unless the**bgpdampening** command is configured with a route map in NSAP address family configuration mode. The following example shows output for the **debugbgpnsapdampening**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
```

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

Table 5: 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.

Field	Description
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 **debugbgpnsapupdates** 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

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

Command Default

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.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
Cisco IOS XE 2.6	This command was integrated into Cisco IOS XE Release 2.6.

Usage Guidelines

The **debugbgpnsapupdates**command is similar to the **debugipbgpupdates**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 **debugbgpnsapupdates**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
```

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

Table 6: 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 bgp vpnv6 unicast

To display Border Gateway Protocol (BGP) virtual private network (VPN) debugging output, use the **debug bgp vpnv6 unicast**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bgp vpnv6 unicast

no debug bgp vpnv6

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

Use the **debug bgp vpnv6 unicast**command to help troubleshoot the BGP VPN.

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. For complete information on debug commands and redirecting debugging output, refer to the Cisco IOS Debug Command Reference, Release 12.4.

Examples

The following example enables BGP debugging output for IPv6 VPN instances:

```
Router# debug bgp vpnv6 unicast
```

debug bri-interface

To display debugging information on ISDN BRI routing activity, use the **debugbri-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**debugbri-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 **debugbri-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 **debugbri-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 **debugbscevent** 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 **bscprotocol-group***number* command.

Examples

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

```
Router# debug bsc event
BSC: Serial2         POLLEE-FSM inp:E_LineFail old_st:CU_Down new_st:TCU_EOFile
BSC: Serial2         POLLEE-FSM inp:E_LineFail old_st:CU_Down new_st:TCU_EOFile
BSC: Serial2         POLLEE-FSM inp:E_LineFail old_st:CU_Down new_st:TCU_EOFile
0:04:32: BSC: Serial2 :SDI-rx: 9 bytes
BSC: Serial2         POLLEE-FSM inp:E_RxEtx old_st:CU_Down new_st:TCU_EOFile
0:04:32: BSC: Serial2 :SDI-rx: 5 bytes
BSC: Serial2         POLLEE-FSM inp:E_RxEnq old_st:CU_Down new_st:TCU_EOFile
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 **debugbscpacket**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).

Command Default

The default number of bytes displayed is 20.

Command Modes

Privileged EXEC

Usage Guidelines

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

Examples

The following is sample output from the **debugbscpacket** 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 **debugbstunevents**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 **debugbstunevents** command, messages showing connection establishment and other overall status messages are displayed.

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

Note Also refer to the **debugbscpacket**and**debugbscevent** 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 **debugbstunevents**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 **debugbstunevents**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 **debugbstunpacket** 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).

Command Default

The default number of bytes displayed is 20.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debugbstunpacket** 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 **debugbundleerrors** 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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

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 **debugbundleevents** 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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

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.

debug call-home diagnostic-signature

To enable the debugging of call-home diagnostic signature flags on a device, use the **debug call-home diagnostic-signature** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug call-home diagnostic-signature {**action | all | api | cli | download | event-registration | parsing**}

no debug call-home diagnostic-signature {**action | all | api | cli | download | event-registration | parsing**}

Syntax Description

action	Displays debugging information associated with the execution of any call-home diagnostic signature action defined in the diagnostic signature file.
all	Displays debugging information about all flags associated with the call-home diagnostic signature.
api	Displays debugging information associated with call-home diagnostic signature internal operations or function calls.
cli	Displays debugging information associated with the call-home diagnostic signature to run the CLI commands as part of the diagnostic signature actions.
download	Displays debugging information associated with the downloading of call-home diagnostic signature files from the HTTP/HTTPS servers.
event-registration	Displays debugging information associated with the registration of call-home diagnostic signature events.
parsing	Displays debugging information associated with the parsing of call-home diagnostic-signature files.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.3(2)T	This command was introduced.

Examples

The following is sample output from the **debug call-home diagnostic-signature action** command:

```
Device# debug call-home diagnostic-signature action

Jan 29 10:42:22.698 CST: DS-ACT-TRACE: call_home_ds_eem_cmd_run[969],
cli cmd "show version", expect string "", max run time 20000
Jan 29 10:42:22.726 CST: DS-ACT-TRACE: call_home_ds_cb_rcmd_lkup[501], cmd "show version"
not exist
```

```
Jan 29 10:42:22.726 CST: DS-ACT-TRACE: call_home_ds_cb_rcmd_add[518], cli show version
Jan 29 10:42:22.726 CST: DS-ACT-TRACE: ds_action_element_next_get[918],
CMD "show version" get the next cmd, done type:DONE_NONE
.
.
.
```

The following is sample output from the **debug call-home diagnostic-signature api** command:

```
Device# debug call-home diagnostic-signature api

Jan 29 10:41:24.902 CST: DS-API-TRACE: call_home_all_lock[101], lock callhome and ds mutex
Jan 29 10:41:24.902 CST: DS-API-TRACE: call_home_ds_lock[42], lock call home ds semaphore
Jan 29 10:41:24.902 CST: DS-API-TRACE: call_home_all_unlock[109], unlock callhome and ds
mutex
Jan 29 10:41:24.902 CST: DS-API-TRACE: call_home_ds_unlock[52], unlock call home ds semaphore
.
.
.
```

The following is sample output from the **debug call-home diagnostic-signature cli** command:

```
Device# debug call-home diagnostic-signature cli

Jan 29 10:44:31.402 CST: DS-CLI-TRACE: call_home_ds_eem_cmd_run[981], the first 100 chars
of output cmd: show version
Cisco IOS Software, C1861 Software (C1861-ADVENTERPRISEK9-M), Experimental Version 15.
Jan 29 10:44:31.442 CST: DS-CLI-TRACE: call_home_ds_eem_cmd_run[981], the first 100 chars
of output cmd: show logging
Syslog logging: enabled (0 messages dropped, 3 messages rate-limited, 0 flushes, 0 over
.
.
.
```

The following is sample output from the **debug call-home diagnostic-signature download** command:

```
Device# debug call-home diagnostic-signature download

Jan 29 10:40:11.050 CST: DS-DNLD-TRACE: call_home_ds_update_thread_create[239], Creating a
 download process
Jan 29 10:40:11.054 CST: DS-DNLD-TRACE: ds_download[119],
url is "https://tools-stage.cisco.com/its/service/oddce/services/DDCEService", num_of_ds
is 1
Jan 29 10:40:11.054 CST: DS-DNLD-TRACE: ds_collect_content_prolog_values[370], Collecting
XML content prolog values
Jan 29 10:40:11.054 CST: DS-DNLD-TRACE: ds_collect_content_prolog_values[489], System
Name:CH1861-1
Jan 29 10:40:11.054 CST: DS-DNLD-TRACE: ds_collect_content_prolog_values[494], Unable to
get SNMP contact string
Jan 29 10:40:11.054 CST: DS-DNLD-TRACE: ds_collect_content_epilog_values[550], Collecting
XML content epilog values
Jan 29 10:40:11.054 CST: DS-DNLD-TRACE: ds_collect_request_values[621], Collecting XML DS
request values
.
.
.
```

The following is sample output from the **debug call-home diagnostic-signature event-registration** command:

```
Device# debug call-home diagnostic-signature event-registration

Jan 29 10:40:16.734 CST: DS-REG-TRACE: call_home_ds_event_register[658], register event for
 ds "6030"
Jan 29 10:40:16.734 CST: DS-REG-TRACE: ds_content_event_reg[515], ds "6030"
Jan 29 10:40:16.734 CST: DS-REG-TRACE: ds_event_sig_reg[304], ds "6030", index 0, event
number 1
Jan 29 10:40:16.734 CST: DS-REG-TRACE: call_home_ds_esid_reg[323], ds "6030", index 0, T =
 41, S = 3FCH
Jan 29 10:40:16.738 CST: DS-REG-TRACE: ds_event_sig_reg[343], ds "6030", register callback
 to action
```

```
.
.
.
```

The following is sample output from the **debug call-home diagnostic-signature parsing** command:

```
Device# debug call-home diagnostic-signature parsing

Jan 29 10:40:16.734 CST: DS-PARSE-TRACE: call_home_ds_signature_verify[387], signature
passed verification
Jan 29 10:40:16.734 CST: DS-REG-TRACE: call_home_ds_update_type_chk[3211], update DS: "6030",
 update type NEW
Jan 29 10:40:16.734 CST: DS-REG-TRACE: call_home_ds_content_reparse[3108], reparse ds "6030"
 content
Jan 29 10:40:16.734 CST: DS-PARSE-TRACE: ds_content_var_reparse[2915], reparse ds var in
6030
Jan 29 10:40:16.734 CST: DS-PARSE-TRACE: ds_sys_var_local_queue_init[2860], copy sys var
ds_signature_id
Jan 29 10:40:16.734 CST: DS-PARSE-TRACE: ds_sys_var_local_queue_init[2860], copy sys var
ds_hostname
.
.
.
```

Related Commands

Command	Description
call-home diagnostic-signature	Downloads, installs, and uninstalls diagnostic signature files on a device.
show call-home diagnostic-signature statistics	Displays statistics and attributes of a diagnostic signature file on a device.

debug call-mgmt

To display debugging information for call accounting, including modem and time slot usage, for active and recent calls, use the **debugcall-mgmt** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call-mgmt

no debug call-mgmt

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1	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 after the **debugcall-mgmt** command has been enabled:

```
Router# debug call-mgmt
Call Management debugging is on
Router#
Dec 26 13:57:27.710: msg_to_calls_mgmt: msg type CPM_NEW_CALL_CSM_CONNECT received
Dec 26 13:57:27.714: In actv_c_proc_message,
   access type CPM_INSERT_NEW_CALL,
   call type CPM_ISDN_ANALOG:
     CSM completed connecting a new modem call
.
.
.
Dec 26 13:57:45.906: msg_to_calls_mgmt: msg type CPM_NEW_CALL_ISDN_CONNECT received
Dec 26 13:57:45.906: In actv_c_proc_message,
   access type CPM_INSERT_NEW_CALL,
   call type CPM_ISDN_ANALOG:
     Added a new ISDN analog call to the active-calls list
     CC-Slot#7, DSX1-Ctrlr#17, DS0-Timeslot#1
     Mdm-Slot#1, Mdm-Port#3, TTY#219
.
.
.
Dec 26 13:58:25.682: Call mgmt per minute statistics:
     active list length: 1
     history list length: 3
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 1
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 2
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 3
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 4
```

```
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 5
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 6
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 7
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 8
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 9
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 10
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 11
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 12
Dec 26 13:58:25.682:      0 timeslots active at slot 7, ctrlr 13
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 14
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 15
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 16
Dec 26 13:58:25.686:      1 timeslots active at slot 7, ctrlr 17
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 18
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 19
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 20
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 21
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 22
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 23
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 24
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 25
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 26
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 27
Dec 26 13:58:25.686:      0 timeslots active at slot 7, ctrlr 28
Router# clear int as1/03
Dec 26 13:58:26.538: msg_to_calls_mgmt: msg type CPM_VOICE_CALL_REJ_NO_MOD_AVAIL received
Dec 26 13:58:26.538: In actv_c_proc_message,
   access type CPM_REMOVE_DISC_CALL,
   call type CPM_ISDN_ANALOG:
     Removed a disconnected ISDN analog call
     CC-Slot#7, DSX1-Ctrlr#17, DS0-Timeslot#1
Dec 26 13:58:26.538:      Mdm-Slot#1, Mdm-Port#3, TTY#219
```

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

Table 7: debug call-mgmt Field Descriptions

Field	Description
CPM_NEW_CALL_CSM_CONNECT	Indicates the arrival of a new call.
access type CPM_INSERT_NEW_CALL, call type CPM_ISDN_ANALOG:	Indicates that the new call is an analog ISDN B channel call (either a voice call or a call over an analog modem), rather than a digital (V.110) call.
CC-Slot#7, DSX1-Ctrlr#17, DS0-Timeslot#1 Mdm-Slot#1, Mdm-Port#3, TTY#219	Indicates that the call is connected via the B channel on Serial7/17:1 to the asynchronous modem resource 1/03 (interface async1/03, also known as line tty219).
Dec 26 13:58:25.682: Call mgmt per minute statistics: active list length: 1 history list length: 3	Displays periodic statistics that give the allocation state of each DSX1 interface present in the system, as well as the number of current (active) and recent (history) calls.
Dec 26 13:58:26.538: msg_to_calls_mgmt: msg type CPM_VOICE_CALL_REJ_NO_MOD_ AVAIL received	Indicates that the analog ISDN B channel call has been disassociated from a modem.

Field	Description
access type CPM_REMOVE_DISC_CALL, call type CPM_ISDN_ANALOG: Removed a disconnected ISDN analog call	Indicates that the analog ISDN B channel call has been disconnected.
CC-Slot#7, DSX1-Ctrlr#17, DS0-Timeslot#1 Dec 26 13:58:26.538: Mdm-Slot#1, Mdm-Port#3, TTY#219	Indicates that the call has been disconnected via the B channel on Serial7/17:1 to the asynchronous modem resource 1/03 (interface async1/03, also known as line tty219).

debug call fallback detail

To display details of the call fallback, use the **debugcallfallbackdetail** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call fallback detail

no debug call fallback detail

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(2)XB1	This command was implemented on the Cisco AS5850 platform.
12.2(4)T	This command was implemented on the Cisco 7200 series routers.
12.2(4)T3	This command was implemented on the Cisco 7500 series routers routers.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T.

Usage Guidelines

Every time a call request is received, the **debugcallfallbackdetail** command displays in the command-line interface (CLI) cache lookup and call acceptance/rejection information. Use this command to monitor call requests as they enter the call fallback subsystem.

If you have a large amount of calls in your router, enabling this command can cause delays in your routing functions as the debug statistics are constantly compiled and sent to your terminal. Also, debug messages on your terminal may make for difficult CLI configuring.

Examples

The following example depicts a call coming in to 10.1.1.4 with codec g729r8. Because there is no cache entry for this destination, a probe is sent and values are inserted into the cache. A lookup is performed again, entry is found, and a fallback decision is made to admit the call.

```
Router# debug call fallback detail
When cache is empty:
debug call fallback detail:
2d19h:fb_lookup_cache:10.1.1.4, codec:g729r8
2d19h:fb_lookup_cache:No entry found.
2d19h:fb_check:no entry exists, enqueueing probe info... 10.1.1.4, codec:g729r8
2d19h:fb_main:Got FB_APP_INQ event
```

```
2d19h:fb_main:Dequeued prob info: 10.1.1.4, codec:g729r8
2d19h:fb_lookup_cache:10.1.1.4, codec:g729r8
2d19h:fb_lookup_cache:No entry found.
2d19h:fb_cache_insert:insert:10.1.1.4, codec:g729r8
2d19h:fb_cache_insert:returning entry:10.1.1.4, codec:g729r8
2d19h:fb_initiate_probe:Creating probe... 10.1.1.4, codec:g729r8
2d19h:fb_initiate_probe:Created and started on probe #13, 10.1.1.4, codec:g729r8
2d19h:fb_lookup_cache:10.1.1.4, codec:g729r8
2d19h:fb_lookup_cache:Found entry.
2d19h:fb_check:returned FB_CHECK_TRUE, 10.1.1.4, codec:g729r8
2d19h:fb_main:calling callback function with:TRUE
```

The following example depicts a call coming in to 10.1.1.4 with codec g729r8. A lookup is performed, entry is found, and a fallback decision is made to admit the call.

```
Router# debug call fallback detail
When cache is full:
2d19h:fb_lookup_cache:10.1.1.4, codec:g729r8
2d19h:fb_lookup_cache:Found entry.
2d19h:fb_check:returned FB_CHECK_TRUE, 10.1.1.4, codec:g729r8
2d19h:fb_main:calling callback function with:TRUE
```

debug call fallback probe

To display details of the call fallback probes, use the **debugcallfallbackprobe** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call fallback probe

no debug call fallback probe

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(2)XA	The **callfallback** and **callfallbackreject-cause-code** commands were introduced.
12.2(2)XB1	This command was implemented on the Cisco AS5850 platform.
12.2(4)T	This command was implemented on the Cisco 7200 series routers.
12.2(4)T3	This command was implemented on the Cisco 7500 series routers.

Usage Guidelines

Every time a probe is received, the **debugcallfallbackprobe** command displays in the command-line interface (CLI) network traffic information collected by the probe. Use this command to monitor the network traffic information the probes carry as they enter the call fallback subsystem and log cache entries.

If you have frequent return of probes to your router, enabling this command can cause delays in your routing functions as the debug statistics are constantly compiled and sent to your terminal. Also, debug messages on your terminal may make for difficult CLI configuring.

Examples

The following example depicts a call coming in to 10.1.1.4 and codec type g729r8. Because there is no cache entry for this IP address, a g729r8 probe is initiated. The probe consists of 20 packet returns with an average delay of 43 milliseconds. The "jitter out" is jitter from source to destination router and "jitter in" is jitter from destination to source router. The delay, loss, and Calculated Planning Impairment Factor (ICPIF) values following g113_calc_icpif are the instantaneous values, whereas those values following "New smoothed values" are the values after applying the smoothing with weight 65.

```
Router# debug call fallback probe
```

```
2d19h:fb_initiate_probe:Probe payload is 32
2d19h:fb_main:NumOfRTT=20, RTTSum=120, loss=0, delay=43, jitter in=0, jitter out=0-> 10.1.1.4,
 codec:g729r8
2d19h:g113_calc_icpif(delay (w/codec delay)=43, loss=0, expect_factor=10) Icpif=0
2d19h:fb_main:Probe timer expired, 10.1.1.4, codec:g729r8
2d19h:fb_main:NumOfRTT=20, RTTSum=120, loss=0, delay=43, jitter in=0, jitter out=0-> 10.1.1.4,
 codec:g729r8
2d19h:g113_calc_icpif(delay (w/codec delay)=43, loss=0, expect_factor=10) Icpif=0
2d19h:fb_main:New smoothed values:inst_weight=65, ICPIF=0, Delay=43, Loss=0 -> 10.1.1.4,
codec:g729r8
```

debug call filter detail

To display details of the debug trace inside the generic call filter module (GCFM), use the debug call filter detail command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call filter detail

no debug call filter detail

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(4)T	This command was introduced.

Examples

The following sample output from the **debugcallfilterdetail** command shows the detailed activity of the GCFM, which is the internal module that controls the debug filtering.

```
Router# debug call filter detail
5d18h: gcfm_call_get_hash_address: hashtable index = 345
5d18h: gcfm_call_search_hash:no found
5d18h: gcfm_init_call_record:
5d18h: gcfm_init_percall_matchlist:
5d18h:  === list 1: service_state=2, callp's: 0
5d18h: gcfm_call_get_hash_address: hashtable index = 345
5d18h: gcfm_call_enlist: count before this enlist 0 on 624D6000
5d18h: gcfm_call_enlist: tail is empty guid=C2E4C789-214A-11D4-804C-000A8A389BA8
5d18h: gcfm_call_get_hash_address: hashtable index = 345
5d18h: gcfm_call_search_hash: search requested guid=C2E4C789-214A-11D4-804C-000A8A389BA8
vs the entry guid=C2E4C789-214A-11D4-804C-000A8A389BA8
5d18h: gcfm_call_search_hash: found
5d18h: gcfm_update_percall_condlist_context:
5d18h: gcfm_update_percall_condlist_context: check cond = 2
5d18h: gcfm_copy_match_cond:
5d18h: gcfm_update_cond_through_matchlist:
5d18h: gcfm_check_percond_with_matchlist: check match-list 1
5d18h: gcfm_matchlist_percond_check:
5d18h: gcfm_matchlist_percond_check: check cond=2
5d18h: gcfm_matchlist_percond_check: compare 42300 to configured 42300
5d18h: gcfm_check_cond_tel_number:
5d18h: gcfm_check_cond_tel_number: matched
5d18h: gcfm_matchlist_percond_check: checked result is 1
5d18h: gcfm_is_bitfield_identical:
5d18h: gcfm_update_cond_through_matchlist: service=1, percallmatchlist tag=1,current_status
 = 1, service_filter=0
5d18h: gcfm_percall_notify_condition: not linked call record
```

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

Table 8: debug call filter detail Field Descriptions

Field	Description
5d18h: gcfm_init_percall_matchlist:	Shows that the filtering has been initiated.
5d18h: gcfm_call_enlist: tail is empty guid=C2E4C789-214A-11D4-804C-000A8A389BA8	Shows the global unique identifier (GUID) for the call.
5d18h: gcfm_check_percond_with_matchlist: check match-list 1	Shows which match list is being checked.
5d18h: gcfm_matchlist_percond_check: checked result is 1	Shows that the call matched conditions in match list 1.

Related Commands

Command	Description
debug call filter inout	Displays the debug trace inside the GCFM.
debug condition match-list	Runs a filtered debug on a voice call.
show call filter components	Displays the components used for filtering calls.

debug call filter inout

To display the debug trace inside the generic call filter module (GCFM), use the debug call filter inout command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call filter inout

no debug call filter inout

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(4)T	This command was introduced.

Examples

The following sample output from the **debugcallfilterinout** command shows the incoming and outgoing activity of the GCFM, which is the internal module that controls the debug filtering.

```
Router# debug call filter inout
5d18h: gcfm_generate_guid:
 component ISDN gets guid
5d18h: gcfm_percall_register:
component ISDN
5d18h: gcfm_percall_register: component ISDN  return selected=0
5d18h: gcfm_percall_notify_condition:
 component ISDN for sync=1
5d18h: gcfm_percall_notify_condition: component ISDN successfully selected = 0
5d18h: gcfm_check_percall_status:
 component TGRM
5d18h: gcfm_check_percall_status: component TGRM  return selected=0
5d18h: gcfm_check_percall_status: component TGRM
5d18h: gcfm_check_percall_status: component TGRM  return selected=0
5d18h: gcfm_percall_register:
 component VTSP
5d18h: gcfm_percall_register: component VTSP for return selected value 0
5d18h: gcfm_percall_notify_condition: component VTSP for sync=1
5d18h: gcfm_percall_notify_condition: component VTSP successfully selected = 0
5d18h: gcfm_percall_register: component CCAPI
5d18h: gcfm_percall_register: component CCAPI for return selected value 0
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION  return selected=0
5d18h: gcfm_percall_register: component VOICE-IVR-V2
5d18h: gcfm_percall_register: component VOICE-IVR-V2 for return selected value 0
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component DIAL-PEER
5d18h: gcfm_check_percall_status: component DIAL-PEER  return selected=0
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component DIAL-PEER
```

```
5d18h: gcfm_check_percall_status: component DIAL-PEER  return selected=0
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component DIAL-PEER
5d18h: gcfm_check_percall_status: component DIAL-PEER  return selected=0
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component DIAL-PEER
5d18h: gcfm_check_percall_status: component DIAL-PEER  return selected=0
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component DIAL-PEER
5d18h: gcfm_check_percall_status: component DIAL-PEER  return selected=0
5d18h: gcfm_check_percall_status: component DIAL-PEER
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_percall_register: component CCAPI
5d18h: gcfm_percall_register: component CCAPI for return selected value 0
5d18h: gcfm_percall_register: component VOICE-IVR-V2
5d18h: gcfm_percall_register: component VOICE-IVR-V2 for return selected value 0
5d18h: gcfm_percall_notify_condition: component VOICE-IVR-V2 for sync=1
5d18h: gcfm_percall_notify_condition: component VOICE-
Router#IVR-V2 successfully selected = 1
5d18h: gcfm_percall_register: component H323
5d18h: gcfm_percall_register: component H323 for return selected value 1
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION  return selected=1
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION  return selected=1
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION  return selected=1
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION  return selected=1
5d18h: gcfm_clear_condition:
 component VOICE-IVR-V2
5d18h: gcfm_clear_condition: component VOICE-IVR-V2 successfully
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION  return selected=0
5d18h: gcfm_percall_deregister:
component CCAPI
5d18h: gcfm_percall_deregister:  component CCAPI successfully
5d18h: gcfm_percall_deregister: component H323
5d18h: gcfm_percall_deregister:  component H323 successfully
5d18h: gcfm_percall_deregister: component ISDN
5d18h: gcfm_percall_deregister:  component ISDN successfully
5d18h: gcfm_percall_deregister: component VOICE-IVR-V2
5d18h: gcfm_percall_deregister:  component VOICE-IVR-V2 successfully
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION
5d18h: gcfm_check_percall_status: component NUMBER-TRANSLATION  return selected=0
5d18h: gcfm_percall_deregister: component CCAPI
5d18h: gcfm_percall_deregister:  component CCAPI successfully
5d18h: gcfm_percall_deregister: component VTSP
5d18h: gcfm_percall_deregister:  component VTSP successfully
5d18h: gcfm_percall_deregister: component VOICE-IVR-V2
5d18h: gcfm_terminate_track_guid:
 component VOICE-IVR-V2 terminate, success
5d18h: gcfm_percall_deregister:  component VOICE-IVR-V2 successfully
```

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

Table 9: debug call filter inout Field Descriptions

Field	Description
gcfm_generate_guid:	Shows that a GUID has been generated.
gcfm_percall_register:	Shows components that have been registered for the call.
gcfm_percall_notify_condition:	Shows that a component has been notified of the call.
gcfm_check_percall_status:	Shows the status of a component of the call.

Field	Description
gcfm_percall_register:	Shows that a component has been registered.
gcfm_clear_condition:	Shows that a condition is cleared for a component.
gcfm_percall_deregister:	Shows that a component has been deregistered.
gcfm_terminate_track_guid:	Shows that the router is no longer tracking the GUID.

Related Commands

Command	Description
debug call filter detail	Displays the details of the debug trace inside the GCFM.
debug condition match-list	Runs a filtered debug on a voice call.
show call filter components	Displays the components used for filtering calls.

debug call rsvp-sync events

To display events that occur during Resource Reservation Protocol (RSVP) setup, use the **debugcallrsvp-syncevents** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call rsvp-sync events

no debug call rsvp-sync events

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XI1	This command was introduced.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.
12.2(2)XB1	This command was implemented on the Cisco AS5850.
12.2(11)T	Support for the command was implemented in Cisco AS5850 images.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

It is highly recommended that you log the output from the **debugcallrsvp-syncevents** command to a buffer, rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Examples

The following example shows a portion of sample output for a call initiating RSVP when using the **debugcallrsvp-syncevents** command:

```
00:03:25:  Parameters: localip: 10.19.101.117 :localport: 16660
00:03:25:  Parameters: remoteip: 10.19.101.116 :remoteport: 17568
00:03:25:  QoS Primitive Event for Call id 0x1 : QoS Listen
00:03:25: Lookup to be done on hashkey 0x1 in hash table 0x61FC2498
00:03:25: Hashed entry 0x1 in call table 0x61FC2498
00:03:25: Entry Not found
00:03:25:  Parameters: localip: 10.19.101.117
00:03:25:      remoteip: 10.19.101.116
00:03:25:  QoSpcb : 0x61FC34D8
00:03:25:  Response Status : 0
Starting timer for call with CallId 0x1 for 10000 secs
00:03:25:  Handling QoS Primitive QoS Listen
```

```
00:03:25:  Establishing RSVP RESV state : rsvp_request_reservation()
00:03:25:  For streams from 10.19.101.116:17568 to 10.19.101.117:16660
00:03:25:  RSVP Confirmation required
00:03:25:  QoS Primitive Event for Call id 0x1 : QoS Resv
00:03:25: Lookup to be done on hashkey 0x1 in hash table 0x61FC2498
00:03:25: Hashed entry 0x1 in call table 0x61FC2498
00:03:25:  Initiating RVSP PATH messages to be Sent : reg_invoke_rsvp_advertise_sender()
00:03:25:  Advertizing for streams to 10.19.101.116:17568 from 10.19.101.117:16660
00:03:25:  RESV notification event received is : 2
00:03:25: Received RESVCONFIRM
00:03:25:  RESV CONFIRM message received from 10.19.101.116 for RESV setup from 10.19.101.117
00:03:25:  RESV event received is : 0
00:03:25:  RESV message received from 10.19.101.116:17568 for streams from 10.19.101.117:16660
00:03:25:  RESERVATIONS ESTABLISHED : CallId: 1       Stop timer and notify Session Protocol
 of Success (ie. if notification requested)
00:03:25:  Invoking spQoSresvCallback with Success
```

Related Commands

Command	Description
call rsvp-sync	Enables synchronization between RSVP and the H.323 voice signaling protocol.
call rsvp-sync resv-timer	Sets the timer for RSVP reservation setup.
debug call rsvp-sync func-trace	Displays messages about the software functions called by RSVP synchronization.
show call rsvp-sync conf	Displays the RSVP synchronization configuration.
show call rsvp-sync stats	Displays statistics for calls that attempted RSVP reservation.

debug call rsvp-sync func-trace

To display messages about software functions called by Resource Reservation Protocol (RSVP), use the **debugcallrsvp-syncfunc-trace** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call rsvp-sync func-trace

no debug call rsvp-sync func-trace

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XI1	This command was introduced.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.
12.2(2)XB1	This command was implemented on the Cisco AS5850.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

It is highly recommended that you log the output from the **debugcallrsvp-syncfunc-trace** command to a buffer, rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Examples

The following example shows a portion of sample output for a call initiating RSVP when using the **debugcallrsvp-syncfunc-trace** command in conjunction with the **debugcallrsvp-syncevents**command:

```
00:03:41: Entering Function QoS_Listen
00:03:41: Parameters:localip:10.10.101.116 :localport:17568
00:03:41:remoteip:10.10.101.117 :remoteport:0
00:03:41: Entering Function qos_dequeue_event
00:03:41: Entering Function process_queue_event
00:03:41: QoS Primitive Event for Call id 0x2 :QoS Listen
00:03:41: Entering Function get_pcb
00:03:41: Entering Function hash_tbl_lookup
00:03:41:Lookup to be done on hashkey 0x2 in hash table 0x61FAECD8
00:03:41: Entering Function hash_func
00:03:41:Hashed entry 0x2 in call table 0x61FAECD8
00:03:41:Entry Not found
00:03:41: Entering Function qos_dequeue_pcb
00:03:41: Entering Function qos_initialize_pcb
```

```
00:03:41: Parameters:localip:10.10.101.116
00:03:41:   remoteip:10.10.101.117
00:03:41: QoSpcb :0x61FAFD18
00:03:41: Response Status :0
00:03:41: Entering Function hash_tbl_insert_entry
00:03:41: Entering Function hash_func
00:03:41: Handling QoS Primitive QoS Listen
00:03:41: Entering Function qos_dequeue_hash_port_entry
00:03:41: Entering Function qos_port_tbl_insert_entry
00:03:41: Entering Function hash_func
00:03:41: Doing RSVP Listen :rsvp_add_ip_listen_api()
```

Related Commands

Command	Description
call rsvp-sync	Enables synchronization between RSVP and the H.323 voice signaling protocol.
call rsvp-sync resv-timer	Sets the timer for RSVP reservation setup.
debug call rsvp-sync events	Displays the events that occur during RSVP synchronization.
show call rsvp-sync conf	Displays the RSVP synchronization configuration.
show call rsvp-sync stats	Displays statistics for calls that attempted RSVP reservation.

debug call threshold

To see details of the trigger actions, use the **debugcallthreshold** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call threshold *module*

no debug call threshold

Syntax Description

module	The *module* argument can be one of the following: • **core** --Traces the resource information. • **detail** --Traces for detail information.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XA	This command was introduced.
12.2(4)T	The command was integrated into Cisco IOS Release 12.2(4)T. Support for the Cisco AS5300, Cisco AS5350, and Cisco AS5400 is not included in this release.
12.2(2)XB1	This command was implemented on the Cisco AS5850 platform.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on Cisco 7200 series routers.
12.2(11)T	Support for this command was implemented on Cisco AS5850, Cisco AS5800, Cisco AS5300, Cisco AS5350, and Cisco AS5400 series images.

Examples

```
The following is sample output from the debug call threshold core
 command:
Router# debug call threshold core
RSCCAC Core info debugging is on
```

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

```
Router# debug call threshold detail
All RSCCAC info debugging is on
```

debug call treatment action

To debug the call treatment actions, use the **debugcalltreatmentaction** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug call treatment action

no debug call treatment action

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XA	This command was introduced.
12.2(4)T	The command was integrated into Cisco IOS Release 12.2(4)T.
12.2(2)XB1	This command was implemented on the Cisco AS5850 platform.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on Cisco 7200 series routers.
12.2(11)T	Support for this command was implemented on Cisco AS5850, Cisco AS5800, Cisco AS5300, Cisco AS5350, and Cisco AS5400 series images.

Examples

Debug actions are performed on calls by call treatment. The following sample output shows that call treatment is turned on:

```
Router# debug call treatment action
Call treatment action debugging is on
```

debug callback

To display callback events when the router is using a modem and a chat script to call back on a terminal line, use the **debugcallback** command in privileged EXEC mode. To disable debugging output, use the **no**form of this command.

debug callback

no debug callback

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command is useful for debugging chat scripts on PPP and AppleTalk Remote Access Protocol (ARAP) lines that use callback mechanisms. The output provided by the **debugcallback** command shows you how the call is progressing when used with the **debugppp** or **debugarap** commands.

Examples

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

```
Router# debug callback

TTY7 Callback process initiated, user: exec_test dialstring 123456
TTY7 Callback forced wait = 4 seconds
TTY7 Exec Callback Successful - await exec/autoselect pickup
TTY7: Callback in effect
```

Related Commands

Command	Description
debug cable env	Displays ARAP events.
debug ppp	Displays information on traffic and exchanges in an internetwork implementing the PPP.

debug capf-server

To collect debug information about the CAPF server, use the **debugcapf-server** command in privileged EXEC mode. To disable collection of debug information, use the **no** form of this command.

debug capf-server {**all**| **error**| **events**| **messages**}

no debug capf-server

Syntax Description

all	Collect all CAPF information available.
error	Collect only information about CAPF errors.
events	Collect only information about CAPF status events.
messages	Collect only CAPF system messages.

Command Default

Collection of CAPF debug information is disabled.

Command Modes

Privileged EXEC

Command History

Cisco IOS Release	Modification
12.4(4)XC	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

This command is used with Cisco Unified CallManager Express phone authentication.

Examples

The following example shows debug messages for the CAPF server.

```
Router# debug capf-server all
001891: .Jul 21 18:17:07.014: %IPPHONE-6-UNREGISTER_NORMAL: ephone-1:SEP000E325C9A43
IP:10.10.10.194 So
cket:3 DeviceType:Phone has unregistered normally.
001892: .Jul 21 18:17:20.495: New Connection from phone, socket 1
001893: .Jul 21 18:17:20.495: Created New Handshake Process
001894: .Jul 21 18:17:20.499: SSL Handshake Error -6983
001895: .Jul 21 18:17:21.499: SSL Handshake Error -6983
001896: .Jul 21 18:17:22.555: SSL Handshake Successful
001897: .Jul 21 18:17:22.555: ephone_capf_send_auth_req:
001898: .Jul 21 18:17:22.555: ephone_capf_ssl_write: 12 bytes
001899: .Jul 21 18:17:22.711: ephone_capf_ssl_read: Read 35 bytes
001900: .Jul 21 18:17:22.711: ephone_capf_handle_phone_msg: msgtype 2
001901: .Jul 21 18:17:22.711: ephone_capf_process_auth_res_msg: SEP000E325C9A43 AuthMode 2
```

```
001902: .Jul 21 18:17:22.711: ephone_capf_send_delete_cert_req_msg: SEP000E325C9A43
001903: .Jul 21 18:17:22.711: ephone_capf_ssl_write: 8 bytes
001904: .Jul 21 18:17:23.891: ephone_capf_ssl_read: Read 12 bytes
001905: .Jul 21 18:17:23.891: ephone_capf_handle_phone_msg: msgtype 14
001906: .Jul 21 18:17:23.891: certificate delete successful for SEP000E325C9A43
001907: .Jul 21 18:17:24.695: ephone_capf_release_session: SEP000E325C9A43
001908: .Jul 21 18:17:24.695: ephone_capf_send_end_session_msg: SEP000E325C9A43
001909: .Jul 21 18:17:24.695: ephone_capf_ssl_write: 12 bytes
001910: .Jul 21 18:17:25.095: %IPPHONE-6-REG_ALARM: 22: Name=SEP000E325C9A43 Load=7.2(2.0)
 Last=Rese
t-Reset
001911: .Jul 21 18:17:25.099: %IPPHONE-6-REGISTER: ephone-1:SEP000E325C9A43 IP:10.10.10.194
 Socket:2 De
viceType:Phone has registered.
001912: .Jul 21 18:18:05.171: %IPPHONE-6-UNREGISTER_NORMAL: ephone-1:SEP000E325C9A43
IP:1.1.1.127 So
cket:2 DeviceType:Phone has unregistered normally.
001913: .Jul 21 18:18:18.288: New Connection from phone, socket 1
001914: .Jul 21 18:18:18.288: Created New Handshake Process
001915: .Jul 21 18:18:18.292: SSL Handshake Error -6983
001916: .Jul 21 18:18:19.292: SSL Handshake Error -6983
001917: .Jul 21 18:18:20.348: SSL Handshake Successful
001918: .Jul 21 18:18:20.348: ephone_capf_send_auth_req:
001919: .Jul 21 18:18:20.348: ephone_capf_ssl_write: 12 bytes^Z
001920: .Jul 21 18:18:20.492: ephone_capf_ssl_read: Read 35 bytes
001921: .Jul 21 18:18:20.492: ephone_capf_handle_phone_msg: msgtype 2
001922: .Jul 21 18:18:20.492: ephone_capf_process_auth_res_msg: SEP000E325C9A43 AuthMode 2
001923: .Jul 21 18:18:20.492: ephone_capf_send_PhKeyGenReq_msg: SEP000E325C9A43 KeySize
1024
001924: .Jul 21 18:18:20.492: ephone_capf_ssl_write: 13 bytes
001925: .Jul 21 18:18:20.540: ephone_capf_ssl_read: Read 8 bytes
001926: .Jul 21 18:18:20.540: ephone_capf_handle_phone_msg: msgtype 17
001927: .Jul 21 18:18:20.540: ephone_capf_process_req_in_progress: SEP000E325C9A43 delay
0sh
001928: .Jul 21 18:18:21.924: %SYS-5-CONFIG_I: Configured from console by user1 on console
```

debug cas

To debug channel-associated signaling (CAS) messages and to debug the establishment of a time-division multiplexing (TDM) connection between a DS0 and a digital modem, use the **debugcas** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cas slot *slot number* **port** *port number*

no debug cas slot *slot number* **port** *port number*

Syntax Description

slot *slot number*	Slot and slot number. Valid values are 0 and 1.
port *port number*	Port and port number. Valid values are 0 and 1.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)T	This command was introduced for the Cisco AS5200 and AS5300 platforms.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T and support was added for the Cisco 2600 series and Cisco 3600 series platforms.
12.3(1)	This command was integrated into Cisco IOS Release 12.3(1) and support was added for the Cisco 2600 XM series, Cisco 2691, and Cisco 3700 series platforms.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

When the NM-xCE1T1PRI network module is used with an NM-xDM and a DS0-group is configured under the controller, you can use the **debugcas** command to debug CAS signaling messages and the establishment of a TDM connection between a DS0 and a digital modem. Use the **debugcas** command to identify and troubleshoot call connection problems on a T1/E1 interface. With this command, you can trace the complete sequence of incoming and outgoing calls.

Examples

The following shows an example session to enable debugging CAS and generate troubleshooting output:

```
Router# show debug

Router# debug cas slot 1 port 0

CAS debugging is on
Router#
debug-cas is on at slot(1) dsx1(0)
Router# show debug

CAS debugging is on
```

The following example shows output for the first outgoing call:

```
Router# p 1.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
*Mar 2 00:17:45: dsx1_alloc_cas_channel: channel 0 dsx1_timeslot
1(0/0): TX SEIZURE (ABCD=0001)(0/0): RX SEIZURE_ACK (ABCD=1101)(0/1):
RX_IDLE (ABCD=1001)(0/2): RX_IDLE (ABCD=1001)(0/3): RX_IDLE
(ABCD=1001)(0/4): RX_IDLE (ABCD=1001)(0/5): RX_IDLE (ABCD=1001)(0/6):
RX_IDLE (ABCD=1001)(0/7): RX_IDLE (ABCD=1001)(0/8): RX_IDLE
(ABCD=1001)(0/9): RX_IDLE (ABCD=1001)(0/10): RX_IDLE (ABCD=1001)(0/11):
RX_IDLE (ABCD=1001)(0/12): RX_IDLE (ABCD=1001)(0/13): RX_IDLE
(ABCD=1001)(0/14): RX_IDLE (ABCD=1001)(0/16): RX_IDLE (ABCD=1001)(0/17):
RX_IDLE (ABCD=1001)(0/18): RX_IDLE (ABCD=1001)(0/19): RX_IDLE
(ABCD=1001)(0/20): RX_IDLE (ABCD=1001)(0/21): RX_IDLE
(ABCD=1001).(0/22): RX_IDLE (ABCD=1001)(0/23): RX_IDLE
(ABCD=1001)(0/24): RX_IDLE (ABCD=1001)(0/25): RX_IDLE (ABCD=1001)(0/26):
RX_IDLE (ABCD=1001)(0/27): RX_IDLE (ABCD=1001)(0/28): RX_IDLE
(ABCD=1001)(0/29): RX_IDLE (ABCD=1001)(0/30): RX_IDLE
(ABCD=1001)...(0/0): RX ANSWERED (ABCD=0101).
Success rate is 0 percent (0/5)
Router#
*Mar 2 00:18:13.333: %LINK-3-UPDOWN: Interface Async94, changed state to up
*Mar 2 00:18:13.333: %DIALER-6-BIND: Interface As94 bound to profile Di1
*Mar 2 00:18:14.577: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async94, changed state
 to up
Router# p 1.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 160/180/236 ms
```

The following example shows that the call is cleared on the router:

```
Router# clear int dialer 1
Router#
(0/0): TX IDLE (ABCD=1001)(0/0): RX IDLE (ABCD=1001)
*Mar 2 00:18:28.617: %LINK-5-CHANGED: Interface Async94, changed state to reset
*Mar 2 00:18:28.617: %DIALER-6-UNBIND: Interface As94 unbound from profile Di1
*Mar 2 00:18:29.617: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async94, changed state
 to down
et2-c3745-1#
*Mar 2 00:18:33.617: %LINK-3-UPDOWN: Interface Async94, changed state to down
```

The following example shows a subsequent outbound CAS call:

```
Router# p 1.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
*Mar 2 00:18:40: dsx1_alloc_cas_channel: channel 5 dsx1_timeslot
6(0/5): TX SEIZURE (ABCD=0001)(0/5): RX SEIZURE_ACK
(ABCD=1101).....(0/5): RX ANSWERED (ABCD=0101).
Success rate is 0 percent (0/5)
Router#
*Mar 2 00:19:08.841: %LINK-3-UPDOWN: Interface Async93, changed state to up
*Mar 2 00:19:08.841: %DIALER-6-BIND: Interface As93 bound to profile Di1
```

```
*Mar 2 00:19:10.033: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async93, changed state
 to up
Router# p 1.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 160/167/176
ms
```

The following example shows the call cleared by the switch:

```
Router#
(0/5): TX IDLE (ABCD=1001)(0/5): RX IDLE (ABCD=1001)
*Mar 2 00:19:26.249: %LINK-5-CHANGED: Interface Async93, changed state to reset
*Mar 2 00:19:26.249: %DIALER-6-UNBIND: Interface As93 unbound from profile Di1
*Mar 2 00:19:27.249: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async93, changed state
 to down
Router#
*Mar 2 00:19:31.249: %LINK-3-UPDOWN: Interface Async93, changed state to down
```

The following example shows an incoming CAS call:

```
Router#
(0/0): RX SEIZURE (ABCD=0001)
*Mar 2 00:22:40: dsx1_alloc_cas_channel: channel 0 dsx1_timeslot
1(0/0): TX SEIZURE_ACK (ABCD=1101)(0/0): TX ANSWERED (ABCD=0101)
Router#
*Mar 2 00:23:06.249: %LINK-3-UPDOWN: Interface Async83, changed state to up
*Mar 2 00:23:06.249: %DIALER-6-BIND: Interface As83 bound to profile Di1
*Mar 2 00:23:07.653: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async83, changed state
 to up
```

Related Commands

Command	Description
show debug	Displays information about the types of debugging that are enabled for your router.

debug ccaal2 session

To display the ccaal2 function calls during call setup and teardown, use the **debugccaal2session**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccaal2 session

no debug ccaal2 session

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging for ATM Adaptation Layer type 2 (AAL2) sessions is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)XA	This command was introduced for the Cisco MC3810 series.
12.1(2)T	This command was integrated in Cisco IOS Release 12.1(2)T.
12.2(2)T	Support for this command was implemented on the Cisco 7200 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command when troubleshooting an AAL2 trunk setup or teardown problem.

Examples

The following example shows sample output from the **debugccaal2session**command for a forced shutdown of a voice port:

```
Router# debug ccaal2 session
CCAAL2 Session debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# voice-port 2/0:0
Router(config-voiceport)# shutdown
00:32:45:ccaal2_call_disconnect:peer tag 0
00:32:45:ccaal2_evhandle_call_disconnect:Entered
00:32:45:ccaal2_call_cleanup:freeccb 1, call_disconnected 1
00:32:45:starting incoming timer:Setting accept_incoming to FALSE and
00:32:45:timer 2:(0x622F6270)starts - delay (70000)
00:32:45:ccaal2_call_cleanup:Generating Call record
00:32:45:cause=81 tcause=81    cause_text=unspecified
00:32:45:ccaal2_call_cleanup:ccb 0x63FF1700, vdbPtr 0x62DFF2E0
        freeccb_flag=1, call_disconnected_flag=1
00:32:45:%LINK-3-UPDOWN:Interface recEive and transMit2/0:0(1),
changed state to Administrative Shutdown
```

The following example shows sample output from the **debugccaal2session**command for a trunk setup on a voice port:

```
Router# debug ccaal2 session
Router(config-voiceport)# no shutdown
Router(config-voiceport)#
00:35:28:%LINK-3-UPDOWN:Interface recEive and transMit2/0:0(1),
changed state to up
00:35:35:ccaal2_call_setup_request:Entered
00:35:35:ccaal2_evhandle_call_setup_request:Entered
00:35:35:ccaal2_initialize_ccb:preferred_codec set(-1)(0)
00:35:35:ccaal2_evhandle_call_setup_request:preferred_codec
set(5)(40). VAD is 1
00:35:35:ccaal2_call_setup_trunk:subchannel linking
successfulccaal2_receive:xmitFunc is NULL
00:35:35:ccaal2_caps_ind:PeerTag = 49
00:35:35:     codec(preferred) = 1, fax_rate = 2, vad = 2
00:35:35:     cid = 56, config_bitmask = 258, codec_bytes = 40,
                      signal_type=8
00:35:36:%HTSP-5-UPDOWN:Trunk port(channel) [2/0:0(1)] is up
Router(config-voiceport)#
```

Related Commands

Command	Description
show debug	Shows which debug commands are enabled.

debug cce dp named-db urlfilter

To enable debug information of the Common Classification Engine Data-Plane (CCE DP) URL Filtering Classification module, use the **debugccedpnamed-dburlfilter** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cce dp named-db urlfilter

no debug cce dp named-db urlfilter

Syntax Description

This command has no keywords or arguments.

Command Default

No debugging information is generated for the the CCE DP URL Filtering Classification module.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(15)XZ	This command was introduced.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Examples

The following is sample output from the **debugccedpnamed-dburlfilter**command at the time that a URL request to the untrusted domain www.example.com was made:

```
Router# debug cce dp named-db urlfilter
CCE DP Named DB URLF functionality debugging is on
Router#
*Apr 4 10:38:08.043: CCE* FUNC: cce_dp_named_db_urlf_pkt_classify -- Didn't get token
*Apr 4 10:38:08.043: CCE* FUNC: cce_dp_urlf_truncate_url -- Truncating URL upto script
before sending to the trend for classification
*Apr  4 10:38:08.043: CCE* FUNC: urlf_trend_find_cache_entry -- The host tree in bucket
1248 is empty
*Apr 4 10:38:08.043: CCE* FUNC: cce_dp_named_db_urlf_pkt_classify -- Didn't find in cache
*Apr 4 10:38:08.051: CCE  FUNC: urlf_trend_store_response -- Host node with given domain
name not found.
*Apr  4 10:38:08.051: CCE  FUNC: urlf_trend_store_response -- Create domain type cache
entry.
*Apr  4 10:38:08.051: CCE  FUNC: cache_size_limit_check -- New cache size=73, existing cache
 size=0, cache size limit=131072000
*Apr  4 10:38:08.051: CCE  FUNC: create_domain_cache_entry -- Domain cache entry 0x65EE0ED0
 created.
*Apr  4 10:38:08.051: CCE  FUNC: create_and_insert_domain_cache_entry --
*Apr  4 10:38:08.051: Domain cache entry 0x65EE0ED0 created and inserted into host tree
with root=0x65EE0ED0, root left=0x0, root right=0x0; new node left=0x0, new node right=0x0
*Apr  4 10:38:08.051: CCE  FUNC: cce_dp_named_db_urlf_gen_match_token -- pushing match-info
 token - class 0xC000000E; filter 45; category 21
*Apr  4 10:38:08.051: CCE  FUNC: cce_dp_named_db_urlf_non_pkt_classify -- Class 0x65C5D484
 matched
*Apr  4 17:38:08.051: %URLF-4-URL_BLOCKED: Access denied URL 'http://www.example.com/',
client 1.0.0.118:3056 server 192.168.0.30:8080
```

```
*Apr  4 10:38:08.055: CCE* FUNC: cce_dp_named_db_urlf_pkt_classify -- Didn't get token
*Apr  4 10:38:08.055: CCE  FUNC: cce_dp_named_db_urlf_pkt_classify -- Didn't get token
```

debug ccfrf11 session

To display the ccfrf11 function calls during call setup and teardown, use the **debugccfrf11session** command in privileged EXEC mode. To disable debugging output, use the **no**form of this command.

debug ccfrf11 session

no debug ccfrf11 session

Syntax Description

This command has no keywords or arguments.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)XG	This command was introduced for the Cisco 2600 and Cisco 3600 series routers.
12.0(4)T	This command was integrated into Cisco IOS Release 12.0(4)T.
12.0(7)XK	This command was first supported on the Cisco MC3810 series.
12.1(2)T	Support for this command was implemented in Cisco MC3810 images.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command to display debug information about the various FRF.11 VoFR service provider interface (SPI) functions. Note that this debug command does not display any information regarding the proprietary Cisco switched-VoFR SPI.

This debug is useful only when the session protocol is "frf11-trunk."

Examples

The following is sample output from the **debugccfrf11session** command:

```
Router# debug ccfrf11 session
INCOMING CALL SETUP (port setup for answer-mode):
*Mar  6 18:04:07.693:ccfrf11_process_timers:scb (0x60EB6040) timer (0x60EB6098) expired
*Mar  6 18:04:07.693:Setting accept_incoming to TRUE
*Mar  6 18:04:11.213:ccfrf11_incoming_request:peer tag 800:callingNumber=+2602100,
      calledNumber=+3622110
*Mar  6 18:04:11.213:ccfrf11_initialize_ccb:preffered_codec set(-1)(0)
*Mar  6 18:04:11.213:ccfrf11_evhandle_incoming_call_setup_request:calling +2602100,
      called +3622110 Incoming Tag 800
*Mar  6 18:04:11.217:ccfrf11_caps_ind:PeerTag = 800
*Mar  6 18:04:11.217:     codec(preferred) = 4, fax_rate = 2, vad = 2
*Mar  6 18:04:11.217:     cid = 30, config_bitmask = 0, codec_bytes = 20, signal_type=2
*Mar  6 18:04:11.217:     required_bandwidth 8192
*Mar  6 18:04:11.217:ccfrf11_caps_ind:Bandwidth reservation of 8192 bytes succeeded.
*Mar  6 18:04:11.221:ccfrf11_evhandle_call_connect:Entered
```

```
CALL SETUP (MASTER):
5d22h:ccfrf11_call_setup_request:Entered
5d22h:ccfrf11_evhandle_call_setup_request:Entered
5d22h:ccfrf11_initialize_ccb:preffered_codec set(-1)(0)
5d22h:ccfrf11_evhandle_call_setup_request:preffered_codec set(9)(24)
5d22h:ccfrf11_call_setup_trunk:subchannel linking successful
5d22h:ccfrf11_caps_ind:PeerTag = 810
5d22h:     codec(preferred) = 512, fax_rate = 2, vad = 2
5d22h:     cid = 30, config_bitmask = 1, codec_bytes = 24, signal_type=2
5d22h:     required_bandwidth 6500
5d22h:ccfrf11_caps_ind:Bandwidth reservation of 6500 bytes succeeded.
CALL TEARDOWN:
*Mar  6 18:09:14.805:ccfrf11_call_disconnect:peer tag 0
*Mar  6 18:09:14.805:ccfrf11_evhandle_call_disconnect:Entered
*Mar  6 18:09:14.805:ccfrf11_call_cleanup:freeccb 1, call_disconnected 1
*Mar  6 18:09:14.805:ccfrf11_call_cleanup:Setting accept_incoming to FALSE and starting
       incoming timer
*Mar  6 18:09:14.809:timer 2:(0x60EB6098)starts - delay (70000)
*Mar  6 18:09:14.809:ccfrf11_call_cleanup:Alive timer stopped
*Mar  6 18:09:14.809:timer 1:(0x60F64104) stops
*Mar  6 18:09:14.809:ccfrf11_call_cleanup:Generating Call record
*Mar  6 18:09:14.809:cause=10 tcause=10    cause_text="normal call clearing."
*Mar  6 18:09:14.809:ccfrf11_call_cleanup:Releasing 8192 bytes of reserved bandwidth
*Mar  6 18:09:14.809:ccfrf11_call_cleanup:ccb 0x60F6404C, vdbPtr 0x610DB7A4
       freeccb_flag=1, call_disconnected_flag=1
```

Related Commands

Command	Description
debug call rsvp-sync events	Displays the ccswvoice function calls during call setup and teardown.
debug ccswvoice vofr-session	Displays the ccswvoice function calls during call setup and teardown.
debug vtsp session	Displays the first 10 bytes (including header) of selected VoFR subframes for the interface.

debug cch323

To provide debugging output for various components within the H.323 subsystem, use the **debugcch323commandinprivilegedEXECmode.**To disable debugging output, use the **no**form of this command.

debug cch323 {**all| error| h225| h245| nxe| ras| rawmsg| session**}

no debug cch323

Syntax Description

all	Enables all debug cch323 commands.
error	Traces errors encountered in the H.323 subsystem and can be used to help troubleshoot problems with H.323 calls.
h225	Traces the state transition of the H.225 state machine on the basis of the processed event.
h245	Traces the state transition of the H.245 state machine on the basis of the processed events.
nxe	Displays Annex E events that have been transmitted and received.
ras	Traces the state transition of the Registration, Admission, and Status (RAS) state machine on the basis of the processed events.
rawmsg	Troubleshoots raw message buffer problems.
session	Traces general H.323 events and can be used to troubleshoot H.323 problems.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(6)NA2	The debug cch323 command and the following keywords were introduced: h225, h245, and ras.
12.2(2)XA	The nxe keyword was added.

Release	Modification
12.2(4)T	The following keywords were introduced: all, error, rawmsg, and session. The nxe keyword was integrated into Cisco IOS Release 12.2(4)T on all Cisco H.323 platforms. This command does not support the Cisco access server platforms in this release.
12.2(2)XB1	This command was implemented on the Cisco AS5850.

Usage Guidelines

The debug cch323 Command with the all Keyword

When used with the **debugcch323** command, the **all** keyword provides debug output for various components within the H.323 subsystem.

The **debugcch323** command used with the **all** keyword enables the following **debugcch323** commands:

error	Enables a CCH323 Service Provider Interface (SPI) trace.
h225	Enables an H225 state machine debugging trace.
h245	Enables an H245 state machine debugging trace.
nxe	Enables an Annex E debugging trace.
ras	Enables a RAS state machine debugging trace.
rawmsg	Enables a CCH323 RAWMSG debugging trace.
session	Enables a Session debugging trace.

Caution Using the **debugcch323all** command could slow your system and flood the TTY if there is significant call traffic.

The debug cch323 Command with the error Keyword

When used with the **debugcch323** command, the **error** keyword allows you to trace errors encountered in the H.323 subsystem.

Note There is little or no output from this command when there is a stable H.323 network.

The debug cch323 Command with the h225 Keyword

When used with the **debugcch323** command, the **h225** keyword allows you to trace the state transition of the H.225 state machine on the basis of the processed event.

The definitions of the different states of the H.225 state machine follow:

- H225_IDLE--This is the initial state of the H.225 state machine. The H.225 state machine is in this state before issuing a call setup request (for the outbound IP call case) or when ready to receive an incoming IP call.
- H225_SETUP--This is the call setup state. The state machine changes to this state after sending out a call setup request or after receiving an incoming call indication.
- H225_ALERT--This is the call alerting state. The state machine changes to this state after sending the alerting message or after receiving an alerting message from the peer.
- H225_CALLPROC--This is the call proceeding state.
- H225_ACTIVE--This is the call connected state. In this state, the call is active. The state machine changes to this state after sending the connect message to the peer or after receiving the connect message from the peer.
- H225_WAIT_FOR_ARQ--This is the state in which the H.225 state machine is waiting for the completion of the Admission Request (ARQ) process from the RAS state machine.
- H225_WAIT_FOR_DRQ--This is the state in which the H.225 state machine is waiting for the completion of the Disengage Request (DRQ) process from the RAS state machine.
- H225_WAIT_FOR_H245--This is the state in which the H.225 state machine is waiting for the success or failure from the H.245 state machine.

The definitions of the different events of the H.225 state machine follow:

- H225_EVENT_NONE--There is no event.
- H225_EVENT_ALERT--This event instructs the H.225 state machine to send an alert message to the peer.
- H225_EVENT_ALERT_IND--This event indicates to the H.225 state machine that an alert message arrived from the peer.
- H225_EVENT_CALLPROC--This event instructs the H.225 state machine to send a call proceeding message to the peer.
- H225_EVENT_CALLPROC_IND--This event indicates to the H.225 state machine that a call proceeding message has been received from the peer.
- H225_EVENT_REJECT--This event instructs the H.225 state machine to reject the call setup request from the peer.
- H225_EVENT_REJECT_IND--This event indicates to the H.225 state machine that a call setup request to the peer has been rejected.
- H225_EVENT_RELEASE--This event instructs the H.225 state machine to send a release complete message to the peer.
- H225_EVENT_RELEASE_IND--This event indicates to the H.225 state machine that a release complete message has been received from the peer.
- H225_EVENT_SETUP--This event instructs the H.225 state machine to send a setup message to the peer.
- H225_EVENT_SETUP_IND--This event indicates to the H.225 state machine that a setup message has been received from the peer.

- H225_EVENT_SETUP_CFM--This event instructs the H.225 state machine to send a connect message to the peer.
- H225_EVENT_SETUP_CFM_IND--This event indicates to the H.225 state machine that a connect message arrived from the peer.
- H225_EVENT_RAS_SUCCESS--This event indicates to the H.225 state machine that the pending RAS operation succeeded.
- H225_EVENT_RAS_FAILED--This event indicates to the H.225 state machine that the pending RAS operation failed.
- H225_EVENT_H245_SUCCESS--This event indicates to the H.225 state machine that the pending H.245 operation succeeded.
- H225_EVENT_H245_FAILED--This event indicates to the H.225 state machine that the pending H.245 operation failed.

The debug cch323 Command with the h245 Keyword

When used with the **debugcch323** command, the **h245** keyword allows you to trace the state transition of the H.245 state machine on the basis of the processed event.

The H.245 state machines include the following three state machines:

- Master slave determination (MSD) state machine
- Capability exchange (CAP) state machine
- Open logical channel (OLC) state machine

The state definitions follow:

- H245_MS_NONE--This is the initial state of the MSD state machine.
- H245_MS_WAIT--In this state, an MSD message is sent, and the device is waiting for the reply.
- H245_MS_DONE-- The result is in.
- H245_CAP_NONE--This is the initial state of the CAP state machine.
- H245_CAP_WAIT--In this state, a CAP message is sent, and the device is waiting for the reply.
- H245_CAP_DONE--The result is in.
- H245_OLC_NONE--This is the initial state of the OLC state machine.
- H245_OLC_WAIT--In this state, an OLC message is sent, and the device is waiting for the reply.
- H245_OLC_DONE--The result is in.

The event definitions follow:

- H245_EVENT_MSD--Send MSD message.
- H245_EVENT_MS_CFM--Send MSD acknowledge message.
- H245_EVENT_MS_REJ--Send MSD reject message.
- H245_EVENT_MS_IND--Received MSD message.
- H245_EVENT_CAP--Send CAP message.

- H245_EVENT_CAP_CFM--Send CAP acknowledge message.
- H245_EVENT_CAP_REJ--Send CAP reject message.
- H245_EVENT_CAP_IND--Received CAP message.
- H245_EVENT_OLC--Send OLC message.
- H245_EVENT_OLC_CFM--Send OLC acknowledge message.
- H245_EVENT_OLC_REJ--Send OLC reject message.
- H245_EVENT_OLC_IND--Received OLC message.

The debug cch323 Command with the nxe Keyword

When used with the **debugcch323** command, the **nxe** keyword allows you to display the Annex E events that have been transmitted and received.

The debug cch323 Command with the ras Keyword

When used with the **debugcch323** command, the **ras** keyword allows you to trace the state transition of the RAS state machine based on the processed events.

RAS operates in two state machines. One global state machine controls the overall RAS operation of the gateway. The other state machine is a per-call state machine that controls the active calls.

The definitions of the different states of the RAS state machine follow:

- CCH323_RAS_STATE_NONE--This is the initial state of the RAS state machine.
- CCH323_RAS_STATE_GRQ--The state machine is in the Gatekeeper Request (GRQ) state. In this state, the gateway is discovering a gatekeeper.
- CCH323_RAS_STATE_RRQ--The state machine is in the Registration Request (RRQ) state. In this state, the gateway is registering with a gatekeeper.
- CCH323_RAS_STATE_IDLE--The global state machine is in the idle state.
- CCH323_RAS_STATE_URQ--The state machine is in the Unregistration Request (URQ) state. In this state, the gateway is in the process of unregistering with a gatekeeper.
- CCH323_RAS_STATE_ARQ--The per-call state machine is in the process of admitting a new call.
- CCH323_RAS_STATE_ACTIVE--The per-call state machine is in the call active state.
- CCH323_RAS_STATE_DRQ--The per-call state machine is in the process of disengaging an active call.

The definitions of the different events of the RAS state machine follow:

- CCH323_RAS_EVENT_NONE--Nothing.
- CCH323_RAS_EVENT_GWUP--Gateway is coming up.
- CCH323_RAS_EVENT_GWDWN--Gateway is going down.
- CCH323_RAS_EVENT_NEWCALL--New call.
- CCH323_RAS_EVENT_CALLDISC--Call disconnect.
- CCH323_RAS_EVENT_GCF--Received Gatekeeper Confirmation (GCF).
- CCH323_RAS_EVENT_GRJ--Received Gatekeeper Rejection (GRJ).

- CCH323_RAS_EVENT_ACF--Received Admission Confirmation (ACF).
- CCH323_RAS_EVENT_ARJ--Received Admission Reject (ARJ).
- CCH323_RAS_EVENT_SEND_RRQ--Send Registration Request (RRQ).
- CCH323_RAS_EVENT_RCF--Received Registration Confirmation (RCF).
- CCH323_RAS_EVENT_RRJ--Received Registration Rejection (RRJ).
- CCH323_RAS_EVENT_SEND_URQ--Send Unregistration Request (URQ).
- CCH323_RAS_EVENT_URQ--Received URQ.
- CCH323_RAS_EVENT_UCF--Received Unregister Confirmation (UCF).
- CCH323_RAS_EVENT_SEND_UCF--Send UCF.
- CCH323_RAS_EVENT_URJ--Received Unregister Reject (URJ).
- CCH323_RAS_EVENT_BCF--Received Bandwidth Confirm (BCF).
- CCH323_RAS_EVENT_BRJ--Received Bandwidth Rejection (BRJ).
- CCH323_RAS_EVENT_DRQ--Received Disengage Request (DRQ).
- CCH323_RAS_EVENT_DCF--Received Disengage Confirm (DCF).
- CCH323_RAS_EVENT_SEND_DCF--Send DCF.
- CCH323_RAS_EVENT_DRJ--Received Disengage Reject (DRJ).
- CCH323_RAS_EVENT_IRQ--Received Interrupt Request (IRQ).
- CCH323_RAS_EVENT_IRR--Send Information Request (IRR).
- CCH323_RAS_EVENT_TIMEOUT--Message timeout.

The debug cch323 Command with the rawmsg Keyword

When used with the **debugcch323** command, the **rawmsg** keyword allows you to troubleshoot raw message buffer problems.

Caution Using the **debugcch323** command with the **rawmsg** keyword could slow your system and flood the TTY if there is significant call traffic.

The debug cch323 Command with the session Keyword

Used with the **debugcch323** command, the **session** keyword allows you to trace general H.323 events.

Caution Using the **debugcch323session** command could slow your system and flood the TTY if there is significant call traffic.

Examples

Examples

The **debugcch323all** command and keyword combination provides output for the following keywords: **error**, **h225**, **h245**, **nxe**, **ras**, **rawmsg**, and **session**. Examples of output for each keyword follow.

Examples The following is sample output from a typical **debugcch323error** request on a Cisco 3640 router:

```
Router# debug cch323 error
cch323_h225_receiver:received msg of unknown type 5
```

Examples The following is sample output from a typical **debugcch323h225** request on a Cisco 3640 router:

```
Router# debug cch323 h225
20:59:17:Set new event H225_EVENT_SETUP
20:59:17:H225 FSM:received event H225_EVENT_SETUP while at state H225_IDLE
20:59:17:Changing from H225_IDLE state to H225_SETUP state
20:59:17:cch323_h225_receiver:received msg of type SETUPCFM_CHOSEN
20:59:17:H225 FSM:received event H225_EVENT_SETUP_CFM_IND while at state
H225_SETUP
20:59:17:Changing from H225_SETUP state to H225_ACTIVE state
20:59:17:Set new event H225_EVENT_H245_SUCCESS
20:59:17:H225 FSM:received event H225_EVENT_H245_SUCCESS while at state
H225_ACTIVE
20:59:20:Set new event H225_EVENT_RELEASE
20:59:20:H225 FSM:received event H225_EVENT_RELEASE while at state
H225_ACTIVE
20:59:20:Changing from H225_ACTIVE state to H225_WAIT_FOR_DRQ state
20:59:20:Set new event H225_EVENT_RAS_SUCCESS
20:59:20:H225 FSM:received event H225_EVENT_RAS_SUCCESS while at state
H225_WAIT_FOR_DRQ
20:59:20:Changing from H225_WAIT_FOR_DRQ state to H225_IDLE state
```

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

Table 10: debug cch323 h225 Field Descriptions

Field	Description
H225_EVENT_SETUP	This event instructs the H.225 state machine to send a setup message to the peer.
H225_IDLE	The initial state of the H.225 state machine. The H.225 state machine is in this state before issuing a call setup request (for the outbound IP call case) or when ready to receive an incoming IP call.
H225_SETUP	The call setup state. The state machine changes to this state after sending out a call setup request or after receiving an incoming call indication.
SETUPCFM_CHOSEN	The H225 connect message that has been received from a remote H323 endpoint.
H225_EVENT_SETUP_CFM_IND	This event indicates to the H.225 state machine that a connect message arrived from the peer.
H225_ACTIVE	The call connected state. In this state, the call is active. The state machine changes to this state after sending the connect message to the peer or after receiving the connect message from the peer.

Field	Description
H225_EVENT_H425_SUCCESS	This event indicates to the H.225 state machine that the pending H.245 operation succeeded.
H225_EVENT_RELEASE	This event instructs the H.225 state machine to send a release complete message to the peer.
H225_WAIT_FOR_DRQ	The state in which the H.225 state machine is waiting for the completion of the DRQ process from the RAS state machine.
H225_EVENT_RAS_SUCCESS	This event indicates to the H.225 state machine that the pending RAS operation succeeded.
H225 FSM	The finite state machine.

Examples

The following is sample output from a typical **debugcch323h245** request on a Cisco 3640 router:

```
Router# debug cch323 h245
20:58:23:Changing to new event H245_EVENT_MSD
20:58:23:H245 MS FSM:received event H245_EVENT_MSD while at state
H245_MS_NONE
20:58:23:changing from H245_MS_NONE state to H245_MS_WAIT state
20:58:23:Changing to new event H245_EVENT_CAP
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP while at state
H245_CAP_NONE
20:58:23:changing from H245_CAP_NONE state to H245_CAP_WAIT state
20:58:23:cch323_h245_receiver:received msg of type
M_H245_MS_DETERMINE_INDICATION
20:58:23:Changing to new event H245_EVENT_MS_IND
20:58:23:H245 MS FSM:received event H245_EVENT_MS_IND while at state
H245_MS_WAIT
20:58:23:cch323_h245_receiver:received msg of type
M_H245_CAP_TRANSFER_INDICATION
20:58:23:Changing to new event H245_EVENT_CAP_IND
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP_IND while at state
H245_CAP_WAIT
20:58:23:cch323_h245_receiver:received msg of type
M_H245_MS_DETERMINE_CONFIRM
20:58:23:Changing to new event H245_EVENT_MS_CFM
20:58:23:H245 MS FSM:received event H245_EVENT_MS_CFM while at state
H245_MS_WAIT
20:58:23:changing from H245_MS_WAIT state to H245_MS_DONE state
0:58:23:cch323_h245_receiver:received msg of type M_H245_CAP_TRANSFER_CONFIRM
20:58:23:Changing to new event H245_EVENT_CAP_CFM
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP_CFM while at state
H245_CAP_WAIT
20:58:23:changing from H245_CAP_WAIT state to H245_CAP_DONE state
20:58:23:Changing to new event H245_EVENT_OLC
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC while at state
H245_OLC_NONE
20:58:23:changing from H245_OLC_NONE state to H245_OLC_WAIT state
20:58:23:cch323_h245_receiver:received msg of type
M_H245_UCHAN_ESTABLISH_INDICATION
20:58:23:Changing to new event H245_EVENT_OLC_IND
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC_IND while at state
H245_OLC_WAIT
20:58:23:cch323_h245_receiver:received msg of type M_H245_UCHAN_ESTAB_ACK
20:58:23:Changing to new event H245_EVENT_OLC_CFM
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC_CFM while at state
```

```
H245_OLC_WAIT
20:58:23:changing from H245_OLC_WAIT state to H245_OLC_DONE state
```

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

Table 11: debug cch323 h245 Field Descriptions

Field	Description
H245_EVENT_MSD	Send MSD event message to the state machine.
H245 MS FSM	An H225 master slave determination finite state machine.
H245_MS_NONE	The initial state of the MSD state machine.
H245_MS_WAIT	In this state, a MSD message is sent, and the device is waiting for the reply.
H245_EVENT_CAP	Send CAP event message.
H245 CAP FSM	This is the H245 terminal CAP finite state machine.
H245_CAP_NONE	The initial state of the CAP state machine.
H245_CAP_WAIT	In this state, a CAP message is sent, and the device is waiting for the reply.
M_H245_MS_DETERMINE _INDICATION	The MSD message that has been received by an H245 terminal from a remote H323 endpoint.
H245_EVENT_MS_IND	Received MSD event message.
M_H245_CAP_TRANSFER_INDICATION	A CAP message that has been received by the H245 terminal from an H323 remote endpoint.
H245_EVENT_CAP_IND	Received CAP event message.
M_H245_MS_DETERMINE_CONFIRM	A confirmation message that the H245 master slave termination message was sent.
H245_EVENT_MS_CFM	Send MSD acknowledge event message.
H245_MS_DONE	The result is in.
M_H245_CAP_TRANSFER_CONFIRM	An indication to the H245 terminal that the CAP message was sent.
H245_EVENT_CAP_CFM	Send CAP acknowledge event message.
H245_CAP_DONE	The result is in.
H245_EVENT_OLC	Send OLC event message.

Field	Description
H245_OLC_NONE	The initial state of the OLC state machine.
H245_OLC_WAIT	In this state, an OLC message is sent, and the device is waiting for the reply.
M_H245_UCHAN_ESTABLISH_INDICATION	The OLC message received by an H245 terminal from a remote H323 endpoint.
H245_EVENT_OLC_IND	Received OLC event message.
M_H245_UCHAN_ESTAB_ACK	The OLC message acknowledgment received by an H245 terminal from a remote H323 endpoint.
H245_EVENT_OLC_CFM	Send OLC acknowledge event message.
H245 OLC FSM	The OLC finite state machine of the H245 terminal.
H245_EVENT_OLC_CFM	Send OLC acknowledge event message.
H245_OLC_DONE	The result is in.

Examples

The following is sample output from a **debugcch323nxe** request:

```
Router# debug cch323 nxe
00:15:54:nxe_handle_usrmsg_to_remote:User Message size is 227
00:15:54:nxe_msg_send_possible:Msg put in the active Q for CRV [3, direction flag 0]
00:15:54:nxe_send_msg:H323chan returns bytes sent=241, the actual len=241, to IPaddr
[0xA4D4A02], Port [2517]
00:15:54:nxe_handle_usrmsg_to_remote:Usr Msg sent for IPaddr [0xA4D4A02], Port [2517], CRV

[3, direction flag 0]
00:15:54:nxe_parse_msg_from_remote:Msg received from IP [0xA4D4A02], Port [2517]
00:15:54:nxe_parse_msg_from_remote:Value of PDU flags = 0x2
00:15:54:nxe_parse_payload:Transport msg type, Payload flag = 0x0
00:15:54:nxe_receive_ack:Ack received for 1 pdus
00:15:54:nxe_receive_ack:Ack received for seqnum=13 from IPAddr [0xA4D4A02], Port [2517]
00:15:54:nxe_parse_msg_from_remote:Msg received from IP [0xA4D4A02], Port [2517]
00:15:54:nxe_parse_msg_from_remote:Value of PDU flags = 0x3
00:15:54:nxe_parse_payload:Static msg type, Payload flag = 0xA0
00:15:54:nxe_parse_x_static:Rx H225 msg from IPaddr [0xA4D4A02], Port [2517], CRV [3,
direction flag 0]
00:15:54:nxe_make_ackmsg:NXE ACK Msg made to ack seqnum=14
00:15:54:nxe_send_msg:H323chan returns bytes sent=16, the actual len=16, to IPaddr
[0xA4D4A02], Port [2517]
00:15:54:nxe_parse_msg_from_remote:Ack sent for Destination IPaddr [0xA4D4A02], Port
[2517]
00:15:54:nxe_parse_msg_from_remote:Msg received from IP [0xA4D4A02], Port [2517]
00:15:54:nxe_parse_msg_from_remote:Value of PDU flags = 0x3
00:15:54:nxe_parse_payload:Static msg type, Payload flag = 0xA0
00:15:54:nxe_parse_x_static:Rx H225 msg from IPaddr [0xA4D4A02], Port [2517], CRV [3,
direction flag 0]
```

Examples

The following is sample output from a typical **debugcch323ras** request on a Cisco 3640 router:

```
Router# debug cch323 ras
20:58:49:Changing to new event CCH323_RAS_EVENT_SEND_RRQ
cch323_run_ras_sm:received event CCH323_RAS_EVENT_SEND_RRQ while at CCH323_RAS_STATE_IDLE
state
cch323_run_ras_sm:changing to CCH323_RAS_STATE_RRQ state
cch323_ras_receiver:received msg of type RCF_CHOSEN
cch323_run_ras_sm:received event CCH323_RAS_EVENT_RCF while at CCH323_RAS_STATE_RRQ state
cch323_run_ras_sm:changing to CCH323_RAS_STATE_IDLE state
20:58:59:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_NEWCALL while at
CCH323_RAS_STATE_IDLE state
20:58:59:cch323_percall_ras_sm:changing to new state CCH323_RAS_STATE_ARQ
cch323_ras_receiver:received msg of type ACF_CHOSEN
20:58:59:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_ACF while at
CCH323_RAS_STATE_ARQ state
20:58:59:cch323_percall_ras_sm:changing to new state
CCH323_RAS_STATE_ACTIVE
20:59:02:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_CALLDISC while
at CCH323_RAS_STATE_ACTIVE state
20:59:02:cch323_percall_ras_sm:changing to new state CCH323_RAS_STATE_DRQ
cch323_ras_receiver:received msg of type DCF_CHOSEN
20:59:02:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_DCF while at
CCH323_RAS_STATE_DRQ state
20:59:02:cch323_percall_ras_sm:changing to new state CCH323_RAS_STATE_IDLE
20:59:04:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_IRR while at
CCH323_RAS_STATE_ACTIVE state
20:59:04:cch323_percall_ras_sm:changing to new state
CCH323_RAS_STATE_ACTIVE
```

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

Table 12: debug cch323 ras Field Descriptions

Field	Description
CCH323_RAS_EVENT_SEND_RRQ	Send RRQ event message.
CCH323_RAS_STATE_IDLE	The global state machine is in the idle state.
CCH323_RAS_STATE_RRQ	The state machine is in the RRQ state. In this state, the gateway is registering with a gatekeeper.
RCF_CHOSEN	A registration confirm message that has been received from a gatekeeper.
CCH323_RAS_EVENT_RCF	Received RCF event message.
CCH323_RAS_EVENT_NEWCALL	New call event.
CCH323_RAS_STATE_ARQ	The per-call state machine is in the process of admitting a new call.
ACF_CHOSEN	ACF message that has been received from a gatekeeper.
CCH323_RAS_EVENT_ACF	Received ACF event message.

Field	Description
CCH323_RAS_STATE_ACTIVE	The per-call state machine is in the call active state.
CCH323_RAS_EVENT_CALLDISC	Call disconnect event message.
CCH323_RAS_STATE_DRQ	The per-call state machine is in the process of disengaging an active call.
DCF_CHOSEN	The disengage confirm message that has been received from a gatekeeper.
CCH323_RAS_EVENT_DCF	Received DCF event message.
CCH323_RAS_EVENT_IRR	Send IRR event message.

Examples

The following is sample output from a typical **debugcch323rawmsg** request on a Cisco 3640 router:

```
Router# debug cch323 rawmsg
00:32:04:cch323_h225_progress_ind:raw message is 4 bytes:1E 02 81 88
00:32:22:cch323_h225_release_ind:raw message is 80 bytes:00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00
00:32:22:cch323_h225_release_notify:raw message is 80 bytes:00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00
```

Examples

Following are two examples of output using the **debugcch323session** command and keyword combination. The first example is for a call setup on an originating gateway. The second example is for a call setup on a terminating gateway.

The following is sample output from a typical **debugcch323session** request for a call setup on an originating gateway:

```
Router# debug cch323 session
00:33:49:cch323_call_setup:gw_id=1, callID=16
00:33:49:cch323_get_new_ccb:ccb (0x81D12D2C) is in use
00:33:49:cch323_call_setup:inserted ccb
cch323_get_peer_info:faxrate[21]proto[2]bitmask[10002]t38_inhibit[0]global_fax[0]
00:33:49:Not using Voice Class Codec
00:33:49:cch323_get_peer_info:preffered_codec set to G729IETF with Bytes = 20
00:33:49:cch323_get_peer_info:peer:81FC0D14, peer->voice_peer_tag:12D, ccb:81D12D2C
00:33:49:Call_setup Playout Mode:0,Init 60, Min 40, Max 200
00:33:49:No account/pin number available
00:33:49:cch323_call_setup_normal:for callID 10
00:33:49:timer (0x81D130D4)starts - delay (15000)
00:33:49:cch323_ct_main:SOCK 1 Event 0x1
00:33:49:timer(0x81D130D4) stops
00:33:49:Near-end Pref Codecs = G.729 IETF
00:33:49: generic_open_logical_channel:codec is g729
00:33:49:cch323_generic_open_logical_channel:Filling in qosCapability field to 0
00:33:49:timer (0x81D130D4)starts - delay (15000)
00:33:49:cch323_ct_main:SOCK 1 Event 0x1
00:33:49:cch323_ct_main:SOCK 1 Event 0x1
00:33:49:      [1]towner_data=0x81D13C88, len=105, msgPtr=0x81D07608
```

```
00:33:49:cch323_gw_process_read_socket:received msg for H.225
00:33:49:timer(0x81D130D4) stops
00:33:49:timer (0x81D130D4)starts - delay (180000)
00:33:49:Codec:loc(16), rem(16),
Bytes:loc(20), Fwd(20), Rev(20)
00:33:49:cch323_rtp_open_notify:
00:33:50:cch323_ct_main:SOCK 1 Event 0x1
00:33:50:      [1]towner_data=0x81D13C88, len=71, msgPtr=0x81F1F2E0
00:33:50:cch323_gw_process_read_socket:received msg for H.225
00:33:50:cch323_caps_ind:cap_modem_proto:0, cap_modem_codec:0, cap_modem_redundancy:0 payload
 100
00:33:50:cch323_caps_ind:Load DSP with Negotiated codec(16) g729r8, Bytes=20
00:33:50:cch323_caps_ind:set DSP for dtmf-relay = CC_CAP_DTMF_RELAY_INBAND_VOICE
```

The following is sample output from a typical **debugcch323session** request for a call setup on a terminating gateway:

```
Router# debug cch323 session
00:23:27:cch323_ct_main:SOCK 0 Event 0x1
00:23:27:cch323_ct_main:SOCK 1 Event 0x1
00:23:27:      [1]towner_data=0x81F9CA9C, len=179, msgPtr=0x81D15C6C
00:23:27:cch323_gw_process_read_socket:received msg for H.225
00:23:27:cch323_h225_receiver CCB not existing already
00:23:27:cch323_get_new_ccb:ccb (0x81F90184) is in use
00:23:27:cch323_h225_receiver Got a new CCB for call id -2115467564
00:23:27:cch323_h225_setup_ind
00:23:27:Not using Voice Class Codec
00:23:27:cch323_set_peer:peer:81FB3228, peer->voice_peer_tag:12C, ccb:81F90184
00:23:27:Near-end Pref Codecs = G.729 IETF
00:23:27:Codec:loc(16), rem(16),
Bytes:loc(20), Fwd(20), Rev(20)
00:23:27:cch323_build_fastStart_cap_response:Retrieved qosCapability of 0
00:23:27:cch323_build_fastStart_cap_response:In Response Filling in qosCapability field
to 0
00:23:27:Not using Voice Class Codec
```

Related Commands

Command	Description
clear h323 gateway	Clears the H.323 gateway counters.
debug h323-annexg	Displays all pertinent AnnexG messages that have been transmitted and received.
debug voip rawmsg	Displays the raw message owner, length, and pointer.
show h323 gateway	Displays statistics for H.323 gateway messages that have been sent and received and displays the reasons for which H.323 calls have been disconnected.

debug cch323 capacity

To track the call capacity of the gatekeeper, use the **debugcch323capacity** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cch323 capacity

no debug cch323 capacity

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced.

Usage Guidelines

Use the **debugcch323capacity** command to track the maximum and current call capacity values in the Registration, Admission, and Status (RAS) Protocol messages and to debug capacity-related problems while sending RAS messages. This command is entered on the gateway to monitor the call capacity of the gatekeeper.

The command lists the values for current and maximum call capacity provided by the trunk group capacity resource manager if and when the H.323 Service Provider Interface (SPI) requests the information for all or specific groups of circuits.

Examples

The following is sample output from the **debugcch323capacity** command:

```
Router# debug cch323 capacity
Call Capacity Information tracing is enabled
5d00h: cch323_process_carrier_update: Registered = 1,Event = 1,Reason = 1
5d00h: cch323_process_carrier_update: CarrierId = CARRIERA_NEWENGLAND
5d00h: cch323_fill_crm_CallCapacities: Reason = 1, GroupID = CARRIERA_NEWENGLAND
5d00h: Capacity Details:           Maximum Channels in Group:    23
      Max. Voice Calls(In) :     23,     Max. Voice Calls(Out):    23
      Active Voice Calls(In):     5,     Active Voice Calls(Out):    7
      Max. Voice Calls(to GK):   23,     Avail. Voice Calls(to GK):   11
```

The gatekeeper displays this output when trunk groups are added, deleted, or modified or when circuits in a trunk group are deactivated or activated (similar to ISDN layer 2 down/up).

```
5d00h: cch323_process_carrier_update: Registered = 1,Event = 1,Reason = 1
5d00h: cch323_process_carrier_update: CarrierId = CARRIERA_NEWENGLAND
```

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

Table 13: debug cch323 capacity Update Field Descriptions

Field	Description
Registered	Gateway registration: • 0=Gateway is not registered to the gatekeeper • 1=Gateway is registered to the gatekeeper at the time of the change
Event	Carriers updated: • 0=All carriers updated • 1=Single carrier updated
Reason	Reason for the update notification: • 0=CURRENT_CAPACITY_UPDATE • 1=MAX_CAPACITY_UPDATE • 2=BOTH_CAPACITY_UPDATE
CarrierID	ID of the trunk group or carrier to which the change applies.

The gatekeeper displays this output whenever call capacity information is sent to the gatekeeper.

```
5d00h: cch323_fill_crm_CallCapacities: Reason = 1, GroupID = CARRIERA_NEWENGLAND
5d00h: Capacity Details:           Maximum Channels in Group:    23
      Max. Voice Calls(In) :     23,     Max. Voice Calls(Out):    23
      Active Voice Calls(In):     5,     Active Voice Calls(Out):    7
      Max. Voice Calls(to GK):   23,     Avail. Voice Calls(to GK):   11
```

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

Table 14: debug cch323 capacity Call Capacity Field Descriptions

Field	Description
GroupID	The circuit's carrier identification (ID) or trunk group label.
Maximum Channels in Group	Maximum number of physical (or configured) circuits.
Max. Voice Calls(In)	Maximum number of allowed incoming voice and data calls.
Max. Voice Calls(Out)	Maximum number of allowed outgoing voice and data calls.

Field	Description
Active Voice Calls(In)	Current number of active incoming voice and data calls.
Active Voice Calls(Out)	Current number of active outgoing voice and data calls.
Max. Voice Calls(to GK)	Maximum call capacity value to be sent to the gatekeeper in the RAS message.
Avail. Voice Calls(to GK)	Available call capacity value to be sent to the gatekeeper in the RAS message.

Related Commands

Command	Description
endpoint circuit-id h323id	Associates a carrier with a non-Cisco endpoint.

debug cch323 h225

To provide the trace of the state transition of the H.225 state machine based on the processed events, use the debug cch323 h225 command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cch323 h225

no debug cch323 h225

Syntax Description

This command has no keywords or arguments.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(6)NA2	This command was introduced.
12.2(2)XB1	This command was implemented on the Cisco AS5850.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T.

Usage Guidelines

State Descriptions

The state definitions of the different states of the H.225 state machine are as follows:

- H225_IDLE--This is the initial state of the H.225 state machine. The H.225 state machine is in this state before issuing a call setup request (for the outbound IP call case) or ready to receive an incoming IP call.
- H225_SETUP--This is the call setup state. The state machine transitions to this state after sending out a call setup request, or after the reception of an incoming call indication.
- H225_ALERT--This is the call alerting state. The state machine transitions to this state after sending the alerting message or after the reception of an alerting message from the peer.
- H225_CALLPROC--This is the call proceeding state.
- H225_ACTIVE--This is the Call connected state. In this state, the call is active. The state machine transitions to this state after sending the connect message to the peer or after the reception of the connect message from the peer.
- H225_WAIT_FOR_ARQ--This is the state where the H.225 state machine is waiting for the completion of the ARQ process from the Registration, Admission, and Status Protocol (RAS) state machine.

- H225_WAIT_FOR_DRQ--This is the state where the H.225 state machine is waiting for the completion of the DRQ process from the RAS state machine.
- H225_WAIT_FOR_H245--This is the state where the H.225 state machine is waiting for the success or failure from the H.245 state machine.

Events Description

The event definitions of the different events of the H.225 state machine are as follows:

- H225_EVENT_NONE-- No event.
- H225_EVENT_ALERT--This event indicates the H.225 state machine to send an alerting message to the peer.
- H225_EVENT_ALERT_IND--This event indicates the H.225 state machine that an alerting message is received from the peer.
- H225_EVENT_CALLPROC--This event indicates the H.225 state machine to send a call proceeding message to the peer.
- H225_EVENT_CALLPROC_IND--This event indicates the H.225 state machine that a call proceeding message is received from the peer.
- H225_EVENT_REJECT--This event indicates the H.225 state machine to reject the call setup request from the peer.
- H225_EVENT_REJECT_IND--This event indicates the H.225 state machine that a call setup request to the peer is rejected.
- H225_EVENT_RELEASE--This event indicates the H.225 state machine to send a release complete message to the peer.
- H225_EVENT_RELEASE_IND--This event indicates the H.225 state machine that a release complete message is received from the peer.
- H225_EVENT_SETUP--This event indicates the H.225 state machine to send a setup message to the peer.
- H225_EVENT_SETUP_IND--This event indicates the H.225 state machine that a setup message is received from the peer.
- H225_EVENT_SETUP_CFM--This event indicates the H.225 state machine to send a connect message to the peer.
- H225_EVENT_SETUP_CFM_IND--This event indicates the H.225 state machine that a connect message from the peer.
- H225_EVENT_RAS_SUCCESS--This event indicates the H.225 state machine that the pending RAS operation is successful.
- H225_EVENT_RAS_FAILED--This event indicates the H.225 state machine that the pending RAS operation failed.
- H225_EVENT_H245_SUCCESS--This event indicates the H.225 state machine that the pending H.245 operation is successful.
- H225_EVENT_H245_FAILED--This event indicates the H.225 state machine that the pending H.245 operation failed.

Examples

The following is sample output from the **debugcch323h225** command:

```
Router# debug cch323 h225
20:59:17:Set new event H225_EVENT_SETUP
20:59:17:H225 FSM:received event H225_EVENT_SETUP while at state H225_IDLE
20:59:17:Changing from H225_IDLE state to H225_SETUP state
20:59:17:cch323_h225_receiver:received msg of type SETUPCFM_CHOSEN
20:59:17:H225 FSM:received event H225_EVENT_SETUP_CFM_IND while at state
H225_SETUP
20:59:17:Changing from H225_SETUP state to H225_ACTIVE state
20:59:17:Set new event H225_EVENT_H245_SUCCESS
20:59:17:H225 FSM:received event H225_EVENT_H245_SUCCESS while at state
H225_ACTIVE
20:59:20:Set new event H225_EVENT_RELEASE
20:59:20:H225 FSM:received event H225_EVENT_RELEASE while at state
H225_ACTIVE
20:59:20:Changing from H225_ACTIVE state to H225_WAIT_FOR_DRQ state
20:59:20:Set new event H225_EVENT_RAS_SUCCESS
20:59:20:H225 FSM:received event H225_EVENT_RAS_SUCCESS while at state
H225_WAIT_FOR_DRQ
20:59:20:Changing from H225_WAIT_FOR_DRQ state to H225_IDLE state
```

debug cch323 h245

To provide the trace of the state transition of the H.245 state machine based on the processed events, use the debug cch323 h245 command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cch323 h245

no debug cch323 h245

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(6)NA2	This command was introduced.
12.2(2)XB1	This command was implemented on the Cisco AS5850.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T.

Usage Guidelines

The H.245 state machines include the following three state machines:

- Master SlaveDetermination (MSD) state machine
- Capability Exchange (CAP) state machine
- Open Logical Channel (OLC) state machine

State Definitions

The definitions are as follows:

- H245_MS_NONE-- This is the initial state of the master slave determination state machine.
- H245_MS_WAIT--In this state, a Master Slave Determination message is sent, waiting for the reply.
- H245_MS_DONE-- The result is in.
- H245_CAP_NONE--This is the initial state of the capabilities exchange state machine.
- H245_CAP_WAIT--In this state, a cap exchange message is sent, waiting for reply.
- H245_CAP_DONE--The result is in.
- H245_OLC_NONE--This is the initial state of the open logical channel state machine.

- H245_OLC_WAIT: OLC message sent, waiting for reply.
- H245_OLC_DONE: OLC done.

Event definitions

- H245_EVENT_MSD--Send MSD message
- H245_EVENT_MS_CFM--Send MSD acknowledge message
- H245_EVENT_MS_REJ--Send MSD reject message
- H245_EVENT_MS_IND-- Received MSD message
- H245_EVENT_CAP--Send CAP message
- H245_EVENT_CAP_CFM--Send CAP acknowledge message
- H245_EVENT_CAP_REJ--Send CAP reject
- H245_EVENT_CAP_IND--Received CAP message
- H245_EVENT_OLC--Send OLC message
- H245_EVENT_OLC_CFM--Send OLC acknowledge message
- H245_EVENT_OLC_REJ--Send OLC reject message
- H245_EVENT_OLC_IND--Received OLC message

Examples

The following is sample output from the **debugcch323h245** command:

```
Router# debug cch323 h245
20:58:23:Changing to new event H245_EVENT_MSD
20:58:23:H245 MS FSM:received event H245_EVENT_MSD while at state
H245_MS_NONE
20:58:23:changing from H245_MS_NONE state to H245_MS_WAIT state
20:58:23:Changing to new event H245_EVENT_CAP
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP while at state
H245_CAP_NONE
20:58:23:changing from H245_CAP_NONE state to H245_CAP_WAIT state
20:58:23:cch323_h245_receiver:received msg of type
M_H245_MS_DETERMINE_INDICATION
20:58:23:Changing to new event H245_EVENT_MS_IND
20:58:23:H245 MS FSM:received event H245_EVENT_MS_IND while at state
H245_MS_WAIT
20:58:23:cch323_h245_receiver:received msg of type
M_H245_CAP_TRANSFER_INDICATION
20:58:23:Changing to new event H245_EVENT_CAP_IND
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP_IND while at state
H245_CAP_WAIT
20:58:23:cch323_h245_receiver:received msg of type
M_H245_MS_DETERMINE_CONFIRM
20:58:23:Changing to new event H245_EVENT_MS_CFM
20:58:23:H245 MS FSM:received event H245_EVENT_MS_CFM while at state
H245_MS_WAIT
20:58:23:changing from H245_MS_WAIT state to H245_MS_DONE state
0:58:23:cch323_h245_receiver:received msg of type M_H245_CAP_TRANSFER_CONFIRM
20:58:23:Changing to new event H245_EVENT_CAP_CFM
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP_CFM while at state
H245_CAP_WAIT
20:58:23:changing from H245_CAP_WAIT state to H245_CAP_DONE state
20:58:23:Changing to new event H245_EVENT_OLC
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC while at state
H245_OLC_NONE
20:58:23:changing from H245_OLC_NONE state to H245_OLC_WAIT state
```

```
20:58:23:cch323_h245_receiver:received msg of type
M_H245_UCHAN_ESTABLISH_INDICATION
20:58:23:Changing to new event H245_EVENT_OLC_IND
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC_IND while at state
H245_OLC_WAIT
20:58:23:cch323_h245_receiver:received msg of type M_H245_UCHAN_ESTAB_ACK
20:58:23:Changing to new event H245_EVENT_OLC_CFM
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC_CFM while at state
H245_OLC_WAIT
20:58:23:changing from H245_OLC_WAIT state to H245_OLC_DONE state
```

debug cch323 preauth

To enable diagnostic reporting of authentication, authorization, and accounting (AAA) call preauthentication for H.323 calls, use the **debugcch323preauth** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cch323 preauth

no debug cch323 preauth

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced.

Examples

The following is debugging output for a single H.323 call:

```
Router# debug cch323 preauth
CCH323 preauth tracing is enabled
cch323_is_preauth_reqd is TRUE
Jan 23 18:39:56.393: In cch323_send_preauth_req for preauth_id = -1
Jan 23 18:39:56.393: Entering rpms_proc_print_preauth_req
Jan 23 18:39:56.393: Request = 0
Jan 23 18:39:56.393: Preauth id = 86514
Jan 23 18:39:56.393: EndPt Type  = 1
Jan 23 18:39:56.393: EndPt = 192.168.81.102
Jan 23 18:39:56.393: Resource Service = 1
Jan 23 18:39:56.393: Call_origin = answer
Jan 23 18:39:56.393: Call_type = voip
Jan 23 18:39:56.393: Calling_num = 2230001
Jan 23 18:39:56.393: Called_num = 1#1130001
Jan 23 18:39:56.393: Protocol  = 0
Jan 23 18:39:56.393: cch323_insert_preauth_tree:Created node with preauth_id = 86514 ,ccb
6852D5BC , node 651F87FC
Jan 23 18:39:56.393:rpms_proc_create_node:Created node with preauth_id = 86514
Jan 23 18:39:56.393:rpms_proc_send_aaa_req:uid got is 466725
Jan 23 18:39:56.397:rpms_proc_preauth_response:Context is for preauth_id 86514, aaa_uid
466725
Jan 23 18:39:56.397: Entering Function cch323_rpms_proc_callback_func
Jan 23 18:39:56.397:cch323_rpms_proc_callback_func:PREAUTH_SUCCESS for preauth id 86514
aaa_uid 466725 auth_serv 1688218168
Jan 23 18:39:56.397:rpms_proc_preauth_response:Deleting Tree node for preauth id 86514 uid
 466725
Jan 23 18:39:56.397:cch323_get_ccb_and_delete_from_preauth_tree:Preauth_id=86514
cch323_get_ccb_and_delete_from_preauth_tree:651F87FC node and 6852D5BC ccb
```

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

Table 15: debug cch323 preauth Field Descriptions

Field	Description
Request	Request Type--0 for preauthentication, 1 for disconnect.
Preauth id	Identifier for the preauthentication request.
EndPt Type	Call Origin End Point Type--1 for IP address, 2 for IZCT value.
EndPt	Call Origin End Point Value--An IP address or IZCT value.
Resource Service	Resource Service Type--1 for Reservation, 2 for Query.
Call_origin	Answer.
Call_type	Voice over IP (VoIP).
Calling_num	Calling Party Number (CLID).
Called_num	Called Party Number (DNIS).
Protocol	0 for H.323, 1 for SIP.
function reports	Various identifiers and status reports for executed functions.

debug cch323 ras

To provide the trace of the state transition of the Registration, Admission, and Status (RAS) Protocol state machine based on the processed events, use the **debugcch323ras** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cch323 ras

no debug cch323 ras

Syntax Description

This command has no keywords or arguments.

Command Default

Disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(6)NA2	This command was introduced.
12.2(2)XB1	This command was implemented on the Cisco AS5850.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T.

Usage Guidelines

RAS operates in two state machines. One global state machine controls the overall RAS operation of the Gateway. The other state machine is a per call state machine that controls the active calls.

State definitions

The state definitions of the different states of the RAS state machine follow:

- CCH323_RAS_STATE_NONE--This is the initial state of the RAS state machine.
- CCH323_RAS_STATE_GRQ--The state machine is in the Gatekeeper Request (GRQ) state. In this state, the gateway is in the process of discovering a gatekeeper.
- CCH323_RAS_STATE_RRQ--The state machine is in the Registration Request (RRQ) state. In this state, the gateway is in the process of registering with a gatekeeper.
- CCH323_RAS_STATE_IDLE--The global state machine is in the idle state.
- CCH323_RAS_STATE_URQ--The state machine is in the Unregistration Request (URQ) state. In this state, the gateway is in the process of unregistering with a gatekeeper.
- CCH323_RAS_STATE_ARQ--The per call state machine is in the process of admitting a new call.
- CCH323_RAS_STATE_ACTIVE--The per call state machine is in the call active state.

- CCH323_RAS_STATE_DRQ--The per call state machine is in the process of disengaging an active call.

Event Definitions

These are the event definitions of the different states of the RAS state machine:

- CCH323_RAS_EVENT_NONE--Nothing.
- CCH323_RAS_EVENT_GWUP--Gateway is coming up.
- CCH323_RAS_EVENT_GWDWN--Gateway is going down.
- CCH323_RAS_EVENT_NEWCALL--New call.
- CCH323_RAS_EVENT_CALLDISC--Call disconnect.
- CCH323_RAS_EVENT_GCF--Received Gatekeeper Confirmation (GCF).
- CCH323_RAS_EVENT_GRJ--Received Gatekeeper Rejection (GRJ).
- CCH323_RAS_EVENT_ACF--Received Admission Confirmation (ACF).
- CCH323_RAS_EVENT_ARJ--Received Admission Rejection (ARJ).
- CCH323_RAS_EVENT_SEND_RRQ--Send Registration Request (RRQ).
- CCH323_RAS_EVENT_RCF--Received Registration Confirmation (RCF).
- CCH323_RAS_EVENT_RRJ--Received Registration Rejection (RRJ).
- CCH323_RAS_EVENT_SEND_URQ--Send URQ.
- CCH323_RAS_EVENT_URQ--Received URQ.
- CCH323_RAS_EVENT_UCF--Received Unregister Confirmation (UCF).
- CCH323_RAS_EVENT_SEND_UCF--Send Unregister Confirmation (UCF).
- CCH323_RAS_EVENT_URJ--Received Unregister Reject (URJ).
- CCH323_RAS_EVENT_BCF--Received Bandwidth Confirm (BCF).
- CCH323_RAS_EVENT_BRJ--Received Bandwidth Rejection (BRJ).
- CCH323_RAS_EVENT_DRQ--Received Disengage Request (DRQ).
- CCH323_RAS_EVENT_DCF--Received Disengage Confirm (DCF).
- CCH323_RAS_EVENT_SEND_DCF--Send Disengage Confirm (DCF).
- CCH323_RAS_EVENT_DRJ--Received Disengage Reject (DRJ).
- CCH323_RAS_EVENT_IRQ--Received Interrupt Request (IRQ).
- CCH323_RAS_EVENT_IRR--Send Information Request (IRR).
- CCH323_RAS_EVENT_TIMEOUT--Message timeout.

Examples

The following is sample output from the **debugcch323preauth** command:

```
Router# debug cch323 preauth
20:58:49:Changing to new event CCH323_RAS_EVENT_SEND_RRQ
```

```
cch323_run_ras_sm:received event CCH323_RAS_EVENT_SEND_RRQ while at CCH323_RAS_STATE_IDLE
state
cch323_run_ras_sm:changing to CCH323_RAS_STATE_RRQ state
cch323_ras_receiver:received msg of type RCF_CHOSEN
cch323_run_ras_sm:received event CCH323_RAS_EVENT_RCF while at CCH323_RAS_STATE_RRQ state
cch323_run_ras_sm:changing to CCH323_RAS_STATE_IDLE state
20:58:59:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_NEWCALL while at
CCH323_RAS_STATE_IDLE state
20:58:59:cch323_percall_ras_sm:changing to new state CCH323_RAS_STATE_ARQ
cch323_ras_receiver:received msg of type ACF_CHOSEN
20:58:59:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_ACF while at
CCH323_RAS_STATE_ARQ state
20:58:59:cch323_percall_ras_sm:changing to new state
CCH323_RAS_STATE_ACTIVE
20:59:02:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_CALLDISC while
at CCH323_RAS_STATE_ACTIVE state
20:59:02:cch323_percall_ras_sm:changing to new state CCH323_RAS_STATE_DRQ
cch323_ras_receiver:received msg of type DCF_CHOSEN
20:59:02:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_DCF while at
CCH323_RAS_STATE_DRQ state
20:59:02:cch323_percall_ras_sm:changing to new state CCH323_RAS_STATE_IDLE
20:59:04:cch323_percall_ras_sm:received event CCH323_RAS_EVENT_IRR while at
CCH323_RAS_STATE_ACTIVE state
20:59:04:cch323_percall_ras_sm:changing to new state
CCH323_RAS_STATE_ACTIVE
```

debug cch323 video

To provide debugging output for video components within the H.323 subsystem, use the **debugcch323videocommandinprivilegedEXECmode.**To disable debugging output, use the **no**form of this command.

debug cch323 video

no debug cch323 video

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Cisco IOS Release	Modification
12.4(4)XC	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

Use this command to enable a debugging trace for the video component in an H.323 network.

Examples

Examples

The following is sample output of the debugging log for an originating Cisco Unified CallManager Express (Cisco Unified CME) gateway after the **debugcch323video** command was enabled:

```
Router# show log
Syslog logging: enabled (11 messages dropped, 487 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 1144 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 1084 message lines logged
Log Buffer (6000000 bytes):
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_peer_info: Entry
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_peer_info: Have peer
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_set_pref_codec_list: First preferred
 codec(bytes)=16(20)
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_peer_info: Flow Mode set to
FLOW_THROUGH
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_caps_chn_info: No peer leg setup
 params
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_caps_chn_info: Setting
CCH323_SS_NTFY_VIDEO_INFO
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_set_h323_control_options_outgoing:
h245 sm mode = 8463
```

```
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_set_h323_control_options_outgoing:
h323_ctl=0x20
Jun 13 09:19:42.010: //103030/C7838B198002/H323/cch323_rotary_validate: No peer_ccb available
```

Examples

The following is sample output of the debugging log for a terminating Cisco Unified Survivable Remote Site Telephony (Cisco Unified SRST) gateway after the **debugcch323video** command was enabled:

```
Router# show log
Syslog logging: enabled (11 messages dropped, 466 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 829 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 771 message lines logged
Log Buffer (200000 bytes):
Jun 13 09:19:42.011: //103034/C7838B198002/H323/setup_ind: Receive bearer cap infoXRate 24,
 rateMult 12
Jun 13 09:19:42.011: //103034/C7838B198002/H323/cch323_set_h245_state_mc_mode_incoming:
h245 state m/c mode=0x10F, h323_ctl=0x2F
Jun 13 09:19:42.015: //-1/xxxxxxxxxxxx/H323/cch245_event_handler: callID=103034
Jun 13 09:19:42.019: //-1/xxxxxxxxxxxx/H323/cch245_event_handler: Event CC_EV_H245_SET_MODE:
 data ptr=0x465D5760
Jun 13 09:19:42.019: //-1/xxxxxxxxxxxx/H323/cch323_set_mode: callID=103034, flow Mode=1
spi_mode=0x6
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_do_call_proceeding: set_mode NOT
called yet...saved deferred CALL_PROC
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_h245_connection_sm: state=0, event=0,
 ccb=4461B518, listen state=0
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_process_set_mode: Setting inbound
leg mode flags to 0x10F, flow-mode to FLOW_THROUGH
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_process_set_mode: Sending deferred
CALL_PROC
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_do_call_proceeding: set_mode called
 so we can proceed with CALLPROC
Jun 13 09:19:42.027: //103034/C7838B198002/H323/cch323_h245_connection_sm: state=1, event=2,
 ccb=4461B518, listen state=1
Jun 13 09:19:42.027: //103034/C7838B198002/H323/cch323_send_cap_request: Setting mode to
VIDEO MODE
Jun 13 09:19:42.031: //103034/C7838B198002/H323/cch323_h245_cap_ind: Masks au=0xC data=0x2
 uinp=0x32
```

Related Commands

Command	Description
debug ephone video	Sets video debugging for the Cisco Unified IP phone.
show call active video	Displays call information for SCCP video calls in progress.
show call history video	Displays call history information for SCCP video calls.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ccm-manager

To display debugging information about Cisco CallManager, use the **debugccm-manager** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccm-manager {**backhaul** {**errors**| **events**| **packets**}| **config-download** {**all**| **errors**| **events**| **packets**| **tone**| **xml**}| **errors**| **events**| **music-on-hold** {**errors**| **events**| **packets**}| **packets**}

no debug ccm-manager

Syntax Description

backhaul	Enables debugging of the Cisco CallManager backhaul. The keywords are as follows: • **errors** --Displays Cisco CallManager backhaul errors. • **events** --Displays Cisco CallManager backhaul events. • **packets** --Displays Cisco CallManager packets.
config-download	Enables debugging of the Cisco CallManager configuration download. The keywords are as follows: • **all** --Displays all Cisco CallManager configuration parameters. • **errors** --Displays Cisco CallManager configuration errors. • **events** --Displays Cisco CallManager configuration events. • **packets** --Displays Cisco CallManager configuration packets. • **tone** --Displays Cisco CallManager downloaded custom tones. • **xml** --Displays the Cisco CallManager configuration XML parser.
errors	Displays errors related to Cisco CallManager.
events	Displays Cisco CallManager events, such as when the primary Cisco CallManager server fails and control is switched to the backup Cisco CallManager server.

music-on-hold	Displays music on hold (MOH). The keywords are as follows: • **errors** --Displays MOH errors. • **events** --Displays MOH events. • **packets** --Displays MOH packets.
packets	Displays Cisco CallManager packets.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced for Cisco CallManager Version 3.0 and the Cisco VG200.
12.2(2)XA	This command was implemented on Cisco 2600 series and Cisco 3600 series routers.
12.2(2)XN	Support for enhanced MGCP voice gateway interoperability was added to Cisco CallManager Version 3.1 for the Cisco 2600 series, Cisco 3600 series, and Cisco VG200.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T and implemented on the Cisco IAD2420 series.
12.2(15)XJ	The **tone** keyword was added for the following platforms: Cisco 2610XM, Cisco 611XM, Cisco 2620XM, Cisco 2621XM, Cisco 2650XM, Cisco 2651XM, Cisco 2691, Cisco 3640A, Cisco 3660, Cisco 3725, and Cisco 3745.
12.3(4)T	The **tone** keyword was added.
12.3(14)T	New output was added relating to the SCCP protocol.
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

The following is sample output from the **debugccm-managerevents** command:

```
Router# debug ccm-manager events
*Feb 28 22:56:05.873: cmapp_mgcpapp_go_down: Setting mgc status to NO_RESPONSE
*Feb 28 22:56:05.873: cmapp_host_fsm: New state DOWN for host 0 (172.20.71.38)
*Feb 28 22:56:05.873: cmapp_mgr_process_ev_active_host_failed: Active host 0 (172.20.71.38)
 failed
*Feb 28 22:56:05.873: cmapp_mgr_check_hostlist: Active host is 0 (172.20.71.38)
```

```
*Feb 28 22:56:05.877: cmapp_mgr_switchover: New actv host will be 1 (172.20.71.44)
*Feb 28 22:56:05.877: cmapp_host_fsm: Processing event GO_STANDBY for host 0 (172.20.71.38)
 in state DOWN
*Feb 28 22:56:05.877: cmapp_open_new_link: Open link for [0]:172.20.71.38
*Feb 28 22:56:05.877: cmbh_open_tcp_link: Opening TCP link with Rem IP 172.20.71.38, Local
 IP 172.20.71.19, port 2428
*Feb 28 22:56:05.881: cmapp_open_new_link: Open initiated OK: Host 0 (172.20.71.38),
session_id=8186DEE4
*Feb 28 22:56:05.881: cmapp_start_open_link_tmr: Host 0 (172.20.71.38), tmr 0
*Feb 28 22:56:05.881: cmapp_host_fsm: New state STANDBY_OPENING for host 0 (172.20.71.38)
*Feb 28 22:56:05.881: cmapp_host_fsm: Processing event GO_ACTIVE for host 1 (172.20.71.44)
 in state STANDBY_READY
*Feb 28 22:56:05.885: cmapp_mgr_send_rehome: new addr=172.20.71.44,port=2427
*Feb 28 22:56:05.885: cmapp_host_fsm: New state REGISTERING for host 1 (172.20.71.44)
```

You can use the **debugccm-managerconfig-downloadtone** command to verify the parameters assigned to each locale. The following sample output shows the locale name United Kingdom and lists all the dual-tone parameters for that region:

```
Router# debug ccm-manager config-download tone
00:09:07:
cmapp_prefix_process_tag_tones:
00:09:07: cmapp_process_tag_trkLocaleName: region = United Kingdom
00:09:07: cmapp_process_tag_pulse_ratio: pulse ratio = 40
00:09:07: cmapp_process_tag_dtmf_llevel: low frequency level = 65438
00:09:07: cmapp_process_tag_dtmf_hlevel: high frequency level = 65463
00:09:07: cmapp_process_tag_special_oper: operation = uLaw
00:09:07: cmapp_prefix_process_tag_lpig:
00:09:07: cmapp_process_tag_fxs: ignore LPIG for fxs
00:09:07: cmapp_process_tag_fxo: ignore LPIG for fxo
00:09:07: cmapp_process_tag_digital: ignore LPIG for digital
00:09:07: cmapp_prefix_process_tag_lpog:
00:09:07: cmapp_process_tag_fxs: ignore LPOG for fxsBoth ports are in service
00:09:07: cmapp_process_tag_fxo: ignore LPOG for fxo
00:09:07: cmapp_process_tag_digital: ignore LPOG for digital
00:09:07: cmapp_prefix_process_tag_tonetable_info:
00:09:07:
cmapp_prefix_process_tag_dualtone: TID=[0:CPTONE_BUSY]
00:09:07: cmapp_process_tag_nf: number of frequencies = 1
00:09:07: cmapp_process_tag_dr: direction = 0
00:09:07: cmapp_process_tag_fof: frequency 1 = 400
00:09:07: cmapp_process_tag_fos: frequency 2 = 0
00:09:07: cmapp_process_tag_fot: frequency 3 = 0
00:09:07: cmapp_process_tag_fo4: frequency 4 = 0
00:09:07: cmapp_prefix_process_tag_aof_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 1st = -200
00:09:07: cmapp_process_tag_fxo: amplitude of 1st = -200
00:09:07: cmapp_process_tag_digital: amplitude of 1st = -240
00:09:07: cmapp_prefix_process_tag_aos_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 2nd = 0
00:09:07: cmapp_process_tag_fxo: amplitude of 2nd = 0
00:09:07: cmapp_process_tag_digital: amplitude of 2nd = 0
00:09:07: cmapp_prefix_process_tag_aot_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 3rd = 0
00:09:07: cmapp_process_tag_fxo: amplitude of 3rd = 0
00:09:07: cmapp_process_tag_digital: amplitude of 3rd = 0
00:09:07: cmapp_prefix_process_tag_ao4_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 4th = 0
00:09:07: cmapp_process_tag_fxo: amplitude of 4th = 0
00:09:07: cmapp_process_tag_digital: amplitude of 4th = 0
00:09:07: cmapp_process_tag_ontf: frequency 1 on time = 375
00:09:07: cmapp_process_tag_oftf: frequency 1 off time = 375
00:09:07: cmapp_process_tag_onts: frequency 2 on time = 0
00:09:07: cmapp_process_tag_ofts: frequency 2 off time = 0
00:09:07: cmapp_process_tag_ontt: frequency 3 on time = 0
00:09:07: cmapp_process_tag_oftt: frequency 3 off time = 0
00:09:07: cmapp_process_tag_ont4: frequency 4 on time = 0
00:09:07: cmapp_process_tag_oft4: frequency 4 off time = 0
00:09:07: cmapp_process_tag_fof2: frequency 1 cadence 2 = 0
00:09:07: cmapp_process_tag_fos2: frequency 2 cadence 2 = 0
00:09:07: cmapp_process_tag_fof3: frequency 1 cadence 3 = 0
```

```
00:09:07: cmapp_process_tag_fos3: frequency 2 cadence 3 = 0
00:09:07: cmapp_process_tag_fof4: frequency 1 cadence 4 = 0
00:09:07: cmapp_process_tag_fos4: frequency 2 cadence 4 = 0
00:09:07: cmapp_process_tag_rct1: cadence 1 repeat count = 0
00:09:07: cmapp_process_tag_rct2: cadence 2 repeat count = 0
00:09:07: cmapp_process_tag_rct3: cadence 3 repeat count = 0
00:09:07: cmapp_process_tag_rct4: cadence 4 repeat count = 0
00:09:07:
cmapp_prefix_process_tag_dualtone: TID=[1:CPTONE_RING_BACK]
00:09:07: cmapp_process_tag_nf: number of frequencies = 2
00:09:07: cmapp_process_tag_dr: direction = 0
00:09:07: cmapp_process_tag_fof: frequency 1 = 400
00:09:07: cmapp_process_tag_fos: frequency 2 = 450
00:09:07: cmapp_process_tag_fot: frequency 3 = 0
00:09:07: cmapp_process_tag_fo4: frequency 4 = 0
00:09:07: cmapp_prefix_process_tag_aof_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 1st = -190
00:09:07: cmapp_process_tag_fxo: amplitude of 1st = -190
00:09:07: cmapp_process_tag_digital: amplitude of 1st = -190
00:09:07: cmapp_prefix_process_tag_aos_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 2nd = -190
00:09:07: cmapp_process_tag_fxo: amplitude of 2nd = -190
00:09:07: cmapp_process_tag_digital: amplitude of 2nd = -190
00:09:07: cmapp_prefix_process_tag_aot_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 3rd = 0
00:09:07: cmapp_process_tag_fxo: amplitude of 3rd = 0
00:09:07: cmapp_process_tag_digital: amplitude of 3rd = 0
00:09:07: cmapp_prefix_process_tag_ao4_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 4th = 0
00:09:07: cmapp_process_tag_fxo: amplitude of 4th = 0
00:09:07: cmapp_process_tag_digital: amplitude of 4th = 0
00:09:07: cmapp_process_tag_ontf: frequency 1 on time = 400
00:09:07: cmapp_process_tag_oftf: frequency 1 off time = 200
00:09:07: cmapp_process_tag_onts: frequency 2 on time = 400
00:09:07: cmapp_process_tag_ofts: frequency 2 off time = 2000
00:09:07: cmapp_process_tag_ontt: frequency 3 on time = 0
00:09:07: cmapp_process_tag_oftt: frequency 3 off time = 0
00:09:07: cmapp_process_tag_ont4: frequency 4 on time = 0
00:09:07: cmapp_process_tag_oft4: frequency 4 off time = 0
00:09:07: cmapp_process_tag_fof2: frequency 1 cadence 2 = 0
00:09:07: cmapp_process_tag_fos2: frequency 2 cadence 2 = 0
00:09:07: cmapp_process_tag_fof3: frequency 1 cadence 3 = 0
00:09:07: cmapp_process_tag_fos3: frequency 2 cadence 3 = 0
00:09:07: cmapp_process_tag_fof4: frequency 1 cadence 4 = 0
00:09:07: cmapp_process_tag_fos4: frequency 2 cadence 4 = 0
00:09:07: cmapp_process_tag_rct1: cadence 1 repeat count = 0
00:09:07: cmapp_process_tag_rct2: cadence 2 repeat count = 0
00:09:07: cmapp_process_tag_rct3: cadence 3 repeat count = 0
00:09:07: cmapp_process_tag_rct4: cadence 4 repeat count = 0
00:09:07:
cmapp_prefix_process_tag_dualtone: TID=[2:CPTONE_CONGESTION]
00:09:07: cmapp_process_tag_nf: number of frequencies = 1
00:09:07: cmapp_process_tag_dr: direction = 0
00:09:07: cmapp_process_tag_fof: frequency 1 = 400
00:09:07: cmapp_process_tag_fos: frequency 2 = 0
00:09:07: cmapp_process_tag_fot: frequency 3 = 0
00:09:07: cmapp_process_tag_fo4: frequency 4 = 0
00:09:07: cmapp_prefix_process_tag_aof_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 1st = -200
00:09:07: cmapp_process_tag_fxo: amplitude of 1st = -200
00:09:07: cmapp_process_tag_digital: amplitude of 1st = -200
00:09:07: cmapp_prefix_process_tag_aos_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 2nd = 0
00:09:07: cmapp_process_tag_fxo: amplitude of 2nd = 0
00:09:07: cmapp_process_tag_digital: amplitude of 2nd = 0
00:09:07: cmapp_prefix_process_tag_aot_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 3rd = 0
00:09:07: cmapp_process_tag_fxo: amplitude of 3rd = 0
00:09:07: cmapp_process_tag_digital: amplitude of 3rd = 0
00:09:07: cmapp_prefix_process_tag_ao4_level:
00:09:07: cmapp_process_tag_fxs: amplitude of 4th = 0
00:09:07: cmapp_process_tag_fxo: amplitude of 4th = 0
00:09:07: cmapp_process_tag_digital: amplitude of 4th = 0
```

```
00:09:07: cmapp_process_tag_ontf: frequency 1 on time = 400
00:09:07: cmapp_process_tag_oftf: frequency 1 off time = 350
00:09:07: cmapp_process_tag_onts: frequency 2 on time = 225
00:09:07: cmapp_process_tag_ofts: frequency 2 off time = 525
00:09:07: cmapp_process_tag_ontt: frequency 3 on time = 0
00:09:07: cmapp_process_tag_oftt: frequency 3 off time = 0
00:09:07: cmapp_process_tag_ont4: frequency 4 on time = 0
00:09:07: cmapp_process_tag_oft4: frequency 4 off time = 0
00:09:07: cmapp_process_tag_fof2: frequency 1 cadence 2 = 0
00:09:07: cmapp_process_tag_fos2: frequency 2 cadence 2 = 0
00:09:07: cmapp_process_tag_fof3: frequency 1 cadence 3 = 0
00:09:07: cmapp_process_tag_fos3: frequency 2 cadence 3 = 0
00:09:07: cmapp_process_tag_fof4: frequency 1 cadence 4 = 0
00:09:07: cmapp_process_tag_fos4: frequency 2 cadence 4 = 0
00:09:07: cmapp_process_tag_rct1: cadence 1 repeat count = 0
00:09:07: cmapp_process_tag_rct2: cadence 2 repeat count = 0
00:09:07: cmapp_process_tag_rct3: cadence 3 repeat count = 0
00:09:07: cmapp_process_tag_rct4: cadence 4 repeat count = 0
! end
```

The following is sample output from the **debugccm-managerconfig-downloadall**command for an error case in which the configuration file cannot be accessed for a Skinny Client Control Protocol (SCCP) download:

```
*Jan  9 07:28:33.499: cmapp_xml_process_timer:
*Jan  9 07:28:33.499: cmapp_xml_find_ep_by_name: Checking for ep_name [*]
*Jan  9 07:28:33.499: cmapp_xml_exec_fsm: Endpoint is [*]
*Jan  9 07:28:33.499: cmapp_xml_exec_fsm: endpoint = * state = CMAPP_XML_FILE_DNLD, event
= CMAPP_XML_EVT_FILE_DNLD_TIMER
*Jan  9 07:28:33.499: cmapp_xml_file_retry_timer_expired: state = CMAPP_XML_FILE_DNLD, event
 = CMAPP_XML_EVT_FILE_DNLD_TIMER
*Jan  9 07:29:14.499: cmapp_xml_tftp_download_file: Unable to read file
tftp://10.6.6.31/Router.cisco.com.cnf.xml, rc=-2
*Jan  9 07:29:14.499: cmapp_xml_get_xml_file: Could not read file
tftp://10.6.6.31/Router.cisco.com.cnf.xml, len = 0
*Jan  9 07:29:14.499: cmapp_xml_tftp_download_file: Unable to read file
tftp:///Router.cisco.com.cnf.xml, rc=-2
*Jan  9 07:29:14.499: cmapp_xml_get_xml_file: Could not read file
tftp:///Router.cisco.com.cnf.xml, len = 0
*Jan  9 07:29:14.499: cmapp_xml_tftp_download_file: Unable to read file
tftp:///Router.cisco.com.cnf.xml, rc=-2
*Jan  9 07:29:14.499: cmapp_xml_get_xml_file: Could not read file
tftp:///Router.cisco.com.cnf.xml, len = 0
*Jan  9 07:29:14.499: cmapp_xml_exec_fsm: New state = CMAPP_XML_FILE_DNLD, ep = 6544CFA8
```

The following is sample output from the **debugccm-managerconfig-downloadall**command for a successful SCCP download:

```
*Jan  9 09:44:45.543: cmapp_sccp_config:
*Jan  9 09:44:45.543: cmapp_sccp_reset_curcfg:
*Jan  9 09:44:45.543: cmapp_sccp_init_curcfg:
*Jan  9 09:44:45.543: cmapp_sccp_download_gw_config_file:
*Jan  9 09:44:45.543: cmapp_sccp_get_gw_name:
*Jan  9 09:44:45.543: cmapp_sccp_get_gw_name: XML file name generated->SKIGW0C85226910.cnf.xml
*Jan  9 09:44:45.543: cmapp_sccp_get_xml_file_via_tftp:
*Jan  9 09:44:45.543: cmapp_sccp_tftp_download_file:
*Jan  9 09:44:45.543: cmapp_sccp_tftp_get_file_size:
*Jan  9 09:44:45.563: cmapp_sccp_get_buffer:
*Jan  9 09:44:45.575: cmapp_sccp_tftp_download_file: File
(tftp://10.2.6.101/SKIGW0C85226910.cnf.xml) read 8162 bytes
*Jan  9 09:44:45.575: cmapp_sccp_get_xml_file_via_tftp: Read file
tftp://10.2.6.101/SKIGW0C85226910.cnf.xml, len = 8162
*Jan  9 09:44:45.575: cmapp_parse_gw_xml_file:
*Jan  9 09:44:45.579: cmapp_sccp_gw_chardata_handler: ccm found, priority=0
```

The following lines show the conversion of XML data into router configuration information for the endpoint:

```
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Unit has been set to 1
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Subunit has been set to 0
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 0
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 1
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 2
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 3
```

```
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Subunit has been set to 1
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 0
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Endpoint has been set to 1
*Jan  9 09:44:45.579: cmapp_sccp_gw_start_element_handler: Unit has been set to 2
```

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

Table 16: debug ccm-manager Field Descriptions

Field	Description
nn :*nn* :*nn* :	Timestamp time in hours (military format), minutes, and seconds that indicates when the Cisco CallManager event occurred.
cmapp_ *error message:*	The Cisco CallManager routine in which the error event occurred.
LocaleName	Region name, such as United Kingdom.
low frequency level	DTMF low frequency.
high frequency level	DTMF high frequency.
operation	Special operations, such as uLaw.

Related Commands

Command	Description
show ccm-manager	Displays a list of Cisco CallManager servers, their current status, and their availability.

debug ccsip all

To enable all Session Initiation Protocol (SIP)-related debugging, use the **debugccsipall** command in privileged EXEC mode. To disable debugging output, use the no form of this command.

debug ccsip all

no debug ccsip all

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

12.1(1)T	This command was introduced.
12.1.(3)T	The output of this command was changed.
12.2(2)XA	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB1	This command was implemented on the Cisco AS5850 universal gateway.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on Cisco 7200 series routers.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T. Support for the Cisco AS5300 universal access server, Cisco AS5350, Cisco AS5400, and Cisco AS5850 universal gateway is not included in this release.

Usage Guidelines

The **debugccsipall** command enables the following SIP debug commands:

- **debug ccsip events**
- **debug ccsip error**
- **debug ccsip states**
- **debug ccsip messages**
- **debug ccsip calls**

Examples

The following example displays debug output from one side of the call:

```
Router# debug ccsip all
```

```
All SIP call tracing enabled
Router1#
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_NONE, SUBSTATE_NONE)  to (STATE_IDLE,
 SUBSTATE_NONE)
*Mar  6 14:10:42:  Queued event from SIP SPI : SIPSPI_EV_CC_CALL_SETUP
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_idle_call_setup
*Mar  6 14:10:42:  act_idle_call_setup:Not using Voice Class Codec
*Mar  6 14:10:42: act_idle_call_setup: preferred_codec set[0] type :g711ulaw bytes: 160
*Mar  6 14:10:42:  Queued event from SIP SPI : SIPSPI_EV_CREATE_CONNECTION
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_IDLE, SUBSTATE_NONE)  to (STATE_IDLE,
 SUBSTATE_CONNECTING)
*Mar  6 14:10:42: REQUEST CONNECTION TO IP:166.34.245.231 PORT:5060
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_IDLE, SUBSTATE_CONNECTING)  to
(STATE_IDLE, SUBSTATE_CONNECTING)
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_idle_connection_created
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_idle_connection_created: Connid(1) created to
166.34.245.231:5060, local_port 54113
*Mar  6 14:10:42: sipSPIAddLocalContact
*Mar  6 14:10:42:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_IDLE, SUBSTATE_CONNECTING)  to
(STATE_SENT_INVITE, SUBSTATE_NONE)
*Mar  6 14:10:42: Sent:
INVITE sip:3660210@166.34.245.231;user=phone;phone-context=unknown SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Sat, 06 Mar 1993 19:10:42 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Cisco-Guid: 2881152943-2184249548-0-483039712
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731427042
Contact: <sip:3660110@166.34.245.230:5060;user=phone>
Expires: 180
Content-Type: application/sdp
Content-Length: 137
v=0
o=CiscoSystemsSIP-GW-UserAgent 1212 283 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20208 RTP/AVP 0
*Mar  6 14:10:42: Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Length: 0
*Mar  6 14:10:42: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.231:5060
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_sentinvite_new_message
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:10:42:  Roundtrip delay 4 milliseconds for method INVITE
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_SENT_INVITE, SUBSTATE_NONE)  to
(STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROCEEDING)
*Mar  6 14:10:42: Received:
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Type: application/sdp
```

```
Content-Length: 137
v=0
o=CiscoSystemsSIP-GW-UserAgent 969 7889 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20038 RTP/AVP 0
*Mar  6 14:10:42: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.231:5060
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_recdproc_new_message
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sipSPICheckResponse : Updating session description
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:10:42:  Roundtrip delay 8 milliseconds for method INVITE
*Mar  6 14:10:42: HandleSIP1xxRinging: SDP MediaTypes negotiation successful!
Negotiated Codec      : g711ulaw , bytes :160
Inband Alerting       : 0
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_PROCEEDING)  to (STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_ALERTING)
*Mar  6 14:10:46: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Contact: <sip:3660210@166.34.245.231:5060;user=phone>
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 137
v=0
o=CiscoSystemsSIP-GW-UserAgent 969 7889 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20038 RTP/AVP 0
*Mar  6 14:10:46: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.231:5060
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  act_recdproc_new_message
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sipSPICheckResponse : Updating session description
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:10:46:  Roundtrip delay 3536 milliseconds for method INVITE
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  act_recdproc_new_message: SDP MediaTypes negotiation
 successful!
Negotiated Codec      : g711ulaw , bytes :160
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sipSPIReconnectConnection
*Mar  6 14:10:46:  Queued event from SIP SPI : SIPSPI_EV_RECONNECT_CONNECTION
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  recv_200_OK_for_invite
*Mar  6 14:10:46:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  6 14:10:46: 0x624CFEF8 : State change from (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_ALERTING)  to (STATE_ACTIVE, SUBSTATE_NONE)
*Mar  6 14:10:46: The Call Setup Information is :
        Call Control Block (CCB) : 0x624CFEF8
         State of The Call        : STATE_ACTIVE
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.230
         Source IP Port    (Media): 20208
         Destn  IP Address (Media): 166.34.245.231
         Destn  IP Port    (Media): 20038
         Destn SIP Addr (Control) : 166.34.245.231
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.231
*Mar  6 14:10:46: HandleUdpReconnection: Udp socket connected for fd: 1 with
166.34.245.231:5060
*Mar  6 14:10:46: Sent:
ACK sip:3660210@166.34.245.231:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
```

```
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
Date: Sat, 06 Mar 1993 19:10:42 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Max-Forwards: 6
Content-Type: application/sdp
Content-Length: 137
CSeq: 101 ACK
v=0
o=CiscoSystemsSIP-GW-UserAgent 1212 283 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20208 RTP/AVP 0
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  ccsip_caps_ind
*Mar  6 14:10:46: ccsip_caps_ind: Load DSP with codec (5) g711ulaw, Bytes=160
*Mar  6 14:10:46: ccsip_caps_ind: set DSP for dtmf-relay = CC_CAP_DTMF_RELAY_INBAND_VOICE
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  ccsip_caps_ack
*Mar  6 14:10:50: Received:
BYE sip:3660110@166.34.245.230:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.231:54835
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
To: "3660110" <sip:3660110@166.34.245.230>
Date: Mon, 08 Mar 1993 22:36:44 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 6
Timestamp: 731612207
CSeq: 101 BYE
Content-Length: 0
*Mar  6 14:10:50: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.231:54835
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  act_active_new_message
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sact_active_new_message_request
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  6 14:10:50:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sipSPIInitiateCallDisconnect : Initiate call
disconnect(16) for outgoing call
*Mar  6 14:10:50: 0x624CFEF8 : State change from (STATE_ACTIVE, SUBSTATE_NONE)  to
(STATE_DISCONNECTING, SUBSTATE_NONE)
*Mar  6 14:10:50: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.231:54835
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
To: "3660110" <sip:3660110@166.34.245.230>
Date: Sat, 06 Mar 1993 19:10:50 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Timestamp: 731612207
Content-Length: 0
CSeq: 101 BYE
*Mar  6 14:10:50:  Queued event From SIP SPI to CCAPI/DNS : SIPSPI_EV_CC_CALL_DISCONNECT
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  act_disconnecting_disconnect
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sipSPICallCleanup
*Mar  6 14:10:50:  Queued event from SIP SPI : SIPSPI_EV_CLOSE_CONNECTION
*Mar  6 14:10:50: CLOSE CONNECTION TO CONNID:1
*Mar  6 14:10:50: sipSPIIcpifUpdate :CallState: 4 Playout: 1755 DiscTime:48305031 ConnTime
 48304651
*Mar  6 14:10:50: 0x624CFEF8 : State change from (STATE_DISCONNECTING, SUBSTATE_NONE)  to
(STATE_DEAD, SUBSTATE_NONE)
*Mar  6 14:10:50: The Call Setup Information is :
        Call Control Block (CCB) : 0x624CFEF8
         State of The Call        : STATE_DEAD
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.230
         Source IP Port    (Media): 20208
         Destn  IP Address (Media): 166.34.245.231
         Destn  IP Port    (Media): 20038
         Destn SIP Addr (Control) : 166.34.245.231
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.231
```

```
*Mar  6 14:10:50:
        Disconnect Cause (CC)    : 16
        Disconnect Cause (SIP)   : 200
*Mar  6 14:10:50: udpsock_close_connect: Socket fd: 1 closed for connid 1 with remote port:
 5060
```

The following example displays debut output from the other side of the call:

```
Router# debug ccsip all
All SIP call tracing enabled
3660-2#
*Mar  8 17:36:40: Received:
INVITE sip:3660210@166.34.245.231;user=phone;phone-context=unknown SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Sat, 06 Mar 1993 19:10:42 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Cisco-Guid: 2881152943-2184249548-0-483039712
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731427042
Contact: <sip:3660110@166.34.245.230:5060;user=phone>
Expires: 180
Content-Type: application/sdp
Content-Length: 137
v=0
o=CiscoSystemsSIP-GW-UserAgent 1212 283 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20208 RTP/AVP 0
*Mar  8 17:36:40: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.230:54113
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sipSPISipIncomingCall
*Mar  8 17:36:40: 0x624D8CCC : State change from (STATE_NONE, SUBSTATE_NONE)  to (STATE_IDLE,
 SUBSTATE_NONE)
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  act_idle_new_message
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sact_idle_new_message_invite
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:36:40:  sact_idle_new_message_invite:Not Using Voice Class Codec
*Mar  8 17:36:40: sact_idle_new_message_invite: Preferred codec[0] type: g711ulaw Bytes
:160
*Mar  8 17:36:40: sact_idle_new_message_invite: Media Negotiation successful for an
incoming call
*Mar  8 17:36:40: sact_idle_new_message_invite: Negotiated Codec      : g711ulaw, bytes
:160
Preferred Codec       : g711ulaw, bytes :160
*Mar  8 17:36:40:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:36:40: Num of Contact Locations 1 3660110 166.34.245.230 5060
*Mar  8 17:36:40: 0x624D8CCC : State change from (STATE_IDLE, SUBSTATE_NONE)  to
(STATE_RECD_INVITE, SUBSTATE_RECD_INVITE_CALL_SETUP)
*Mar  8 17:36:40: Sent:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Length: 0
*Mar  8 17:36:40:  Queued event From SIP SPI to CCAPI/DNS : SIPSPI_EV_CC_CALL_PROCEEDING
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  act_recdinvite_proceeding
*Mar  8 17:36:40:  Queued event From SIP SPI to CCAPI/DNS : SIPSPI_EV_CC_CALL_ALERTING
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  ccsip_caps_ind
*Mar  8 17:36:40: ccsip_caps_ind: codec(negotiated) = 5(Bytes 160)
*Mar  8 17:36:40: ccsip_caps_ind: Load DSP with codec (5) g711ulaw, Bytes=160
*Mar  8 17:36:40: ccsip_caps_ind: set DSP for dtmf-relay = CC_CAP_DTMF_RELAY_INBAND_VOICE
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  ccsip_caps_ack
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  act_recdinvite_alerting
```

```
*Mar  8 17:36:40:  180 Ringing with SDP - not likely
*Mar  8 17:36:40:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:36:40: 0x624D8CCC : State change from (STATE_RECD_INVITE,
SUBSTATE_RECD_INVITE_CALL_SETUP)  to (STATE_SENT_ALERTING, SUBSTATE_NONE)
*Mar  8 17:36:40: Sent:
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 137
v=0
o=CiscoSystemsSIP-GW-UserAgent 969 7889 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20038 RTP/AVP 0
*Mar  8 17:36:44:  Queued event From SIP SPI to CCAPI/DNS : SIPSPI_EV_CC_CALL_CONNECT
*Mar  8 17:36:44: CCSIP-SPI-CONTROL:  act_sentalert_connect
*Mar  8 17:36:44: sipSPIAddLocalContact
*Mar  8 17:36:44:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  8 17:36:44: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:36:44: 0x624D8CCC : State change from (STATE_SENT_ALERTING, SUBSTATE_NONE)  to
(STATE_SENT_SUCCESS, SUBSTATE_NONE)
*Mar  8 17:36:44: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Contact: <sip:3660210@166.34.245.231:5060;user=phone>
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 137
v=0
o=CiscoSystemsSIP-GW-UserAgent 969 7889 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20038 RTP/AVP 0
*Mar  8 17:36:44: Received:
ACK sip:3660210@166.34.245.231:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
Date: Sat, 06 Mar 1993 19:10:42 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Max-Forwards: 6
Content-Type: application/sdp
Content-Length: 137
CSeq: 101 ACK
v=0
o=CiscoSystemsSIP-GW-UserAgent 1212 283 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20208 RTP/AVP 0
*Mar  8 17:36:44: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.230:54113
*Mar  8 17:36:44: CCSIP-SPI-CONTROL:  act_sentsucc_new_message
*Mar  8 17:36:44: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:36:44: 0x624D8CCC : State change from (STATE_SENT_SUCCESS, SUBSTATE_NONE)  to
(STATE_ACTIVE, SUBSTATE_NONE)
*Mar  8 17:36:44: The Call Setup Information is :
        Call Control Block (CCB) : 0x624D8CCC
```

```
          State of The Call        : STATE_ACTIVE
          TCP Sockets Used         : NO
          Calling Number           : 3660110
          Called Number            : 3660210
          Negotiated Codec         : g711ulaw
          Source IP Address (Media): 166.34.245.231
          Source IP Port    (Media): 20038
          Destn  IP Address (Media): 166.34.245.230
          Destn  IP Port    (Media): 20208
          Destn SIP Addr (Control) : 166.34.245.230
          Destn SIP Port (Control) : 5060
          Destination Name         : 166.34.245.230
*Mar  8 17:36:47:  Queued event From SIP SPI to CCAPI/DNS : SIPSPI_EV_CC_CALL_DISCONNECT
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  act_active_disconnect
*Mar  8 17:36:47:  Queued event from SIP SPI : SIPSPI_EV_CREATE_CONNECTION
*Mar  8 17:36:47: 0x624D8CCC : State change from (STATE_ACTIVE, SUBSTATE_NONE)  to
(STATE_ACTIVE, SUBSTATE_CONNECTING)
*Mar  8 17:36:47: REQUEST CONNECTION TO IP:166.34.245.230 PORT:5060
*Mar  8 17:36:47: 0x624D8CCC : State change from (STATE_ACTIVE, SUBSTATE_CONNECTING)  to
(STATE_ACTIVE, SUBSTATE_CONNECTING)
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  act_active_connection_created
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sipSPICheckSocketConnection
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sipSPICheckSocketConnection: Connid(1) created to
166.34.245.230:5060, local_port 54835
*Mar  8 17:36:47:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:36:47: 0x624D8CCC : State change from (STATE_ACTIVE, SUBSTATE_CONNECTING)  to
(STATE_DISCONNECTING, SUBSTATE_NONE)
*Mar  8 17:36:47: Sent:
BYE sip:3660110@166.34.245.230:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.231:54835
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
To: "3660110" <sip:3660110@166.34.245.230>
Date: Mon, 08 Mar 1993 22:36:44 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 6
Timestamp: 731612207
CSeq: 101 BYE
Content-Length: 0
*Mar  8 17:36:47: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.231:54835
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
To: "3660110" <sip:3660110@166.34.245.230>
Date: Sat, 06 Mar 1993 19:10:50 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Timestamp: 731612207
Content-Length: 0
CSeq: 101 BYE
*Mar  8 17:36:47: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.230:54113
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  act_disconnecting_new_message
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sact_disconnecting_new_message_response
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:36:47:  Roundtrip delay 4 milliseconds for method BYE
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sipSPICallCleanup
*Mar  8 17:36:47:  Queued event from SIP SPI : SIPSPI_EV_CLOSE_CONNECTION
*Mar  8 17:36:47: CLOSE CONNECTION TO CONNID:1
*Mar  8 17:36:47: sipSPIIcpifUpdate :CallState: 4 Playout: 1265 DiscTime:66820800 ConnTime
 66820420
*Mar  8 17:36:47: 0x624D8CCC : State change from (STATE_DISCONNECTING, SUBSTATE_NONE)  to
(STATE_DEAD, SUBSTATE_NONE)
*Mar  8 17:36:47: The Call Setup Information is :
        Call Control Block (CCB) : 0x624D8CCC
          State of The Call        : STATE_DEAD
          TCP Sockets Used         : NO
          Calling Number           : 3660110
          Called Number            : 3660210
          Negotiated Codec         : g711ulaw
          Source IP Address (Media): 166.34.245.231
          Source IP Port    (Media): 20038
```

```
         Destn  IP Address (Media): 166.34.245.230
         Destn  IP Port    (Media): 20208
         Destn SIP Addr (Control) : 166.34.245.230
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.230
*Mar  8 17:36:47:
        Disconnect Cause (CC)    : 16
        Disconnect Cause (SIP)   : 200
*Mar  8 17:36:47: udpsock_close_connect: Socket fd: 1 closed for connid 1 with remote port:
 5060
```

Related Commands

Command	Description
debug ccsip calls	Shows all SIP SPI call tracing.
debug ccsip error	Shows SIP SPI errors.
debug ccsip events	Shows all SIP SPI events tracing.
debug ccsip info	Shows all SIP SPI message tracing.
debug ccsip states	Shows all SIP SPI state tracing.

debug ccsip calls

To show all Session Initiation Protocol (SIP) Service Provider Interface (SPI) call tracing, use the **debugccsipcalls** command in privileged EXEC mode. To disable debugging output, use the no form of this command.

debug ccsip calls

no debug ccsip calls

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

12.1(1)T	This command was introduced.
12.1(3)T	The output of this command was changed.
12.2(2)XA	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB1	This command was introduced on the Cisco AS5850 universal gateway.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on Cisco 7200 series routers.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T. Support for the Cisco AS5300 universal access server, Cisco AS5350, Cisco AS5400, and Cisco AS5850 universal gateway is not included in this release.

Usage Guidelines

This command traces the SIP call details as they are updated in the SIP call control block.

Examples

The following example displays debug output from one side of the call:

```
Router1# debug ccsip calls
SIP Call statistics tracing is enabled
Router1#
*Mar  6 14:12:33: The Call Setup Information is :
        Call Control Block (CCB) : 0x624D078C
         State of The Call        : STATE_ACTIVE
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.230
```

```
         Source IP Port    (Media): 20644
         Destn  IP Address (Media): 166.34.245.231
         Destn  IP Port    (Media): 20500
         Destn SIP Addr (Control) : 166.34.245.231
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.231
*Mar  6 14:12:40: The Call Setup Information is :
        Call Control Block (CCB) : 0x624D078C
         State of The Call        : STATE_DEAD
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.230
         Source IP Port    (Media): 20644
         Destn  IP Address (Media): 166.34.245.231
         Destn  IP Port    (Media): 20500
         Destn SIP Addr (Control) : 166.34.245.231
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.231
*Mar  6 14:12:40:
        Disconnect Cause (CC)    : 16
        Disconnect Cause (SIP)   : 200
```

The following example displays debug output from the other side of the call:

```
Router2# debug ccsip calls
SIP Call statistics tracing is enabled
Router2#
*Mar  8 17:38:31: The Call Setup Information is :
        Call Control Block (CCB) : 0x624D9560
         State of The Call        : STATE_ACTIVE
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.231
         Source IP Port    (Media): 20500
         Destn  IP Address (Media): 166.34.245.230
         Destn  IP Port    (Media): 20644
         Destn SIP Addr (Control) : 166.34.245.230
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.230
*Mar  8 17:38:38: The Call Setup Information is:
        Call Control Block (CCB) : 0x624D9560
         State of The Call        : STATE_DEAD
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.231
         Source IP Port    (Media): 20500
         Destn  IP Address (Media): 166.34.245.230
         Destn  IP Port    (Media): 20644
         Destn SIP Addr (Control) : 166.34.245.230
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.230
*Mar  8 17:38:38:
        Disconnect Cause (CC)    : 16
        Disconnect Cause (SIP)   : 200
```

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging.
debug ccsip error	Shows SIP SPI errors.
debug ccsip events	Shows all SIP SPI events tracing.

Command	Description
debug ccsip info	Shows all SIP SPI message tracing.
debug ccsip states	Shows all SIP SPI state tracing.

debug ccsip dhcp

To display debugging related information on Session Initiation Protocol (SIP) and Dynamic Host Configuration Protocol (DHCP) interaction, when SIP parameters are provisoned by DHCP, use the **debugccsipdhcp** command in privileged EXEC mode. To disable debugging output, use the no form of this command.

debug ccsip dhcp

no debug ccsip dhcp

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

12.4(22)YB	This command was introduced.
15.0(1)M	This command was integrated in Cisco IOS Release 15.0(1)M.

Usage Guidelines

The debug ccsip dhcp command can be enabled by executing the command itself or by issuing the debug ccsip all command.

Examples

The following example displays debug output from the debug ccsip dhcp command:

```
Router# debug ccsip dhcp
Nov 18 17:20:48.881: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_register_configured_dest_patterns:
No destination patterns to Register
Nov 18 17:20:48.881: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_spi_register_free_rcb: Freeing rcb
Nov 18 17:20:48.881: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_register_reset_dns_cache:
CCSIP_REGISTER:: Primary registrar DNS resolved addr reset
Nov 18 17:21:00.965: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/config_credential_trigger_reg: Query
DHCP for provisioned info upon credential dhcp config
Nov 18 17:21:00.965: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/sipua_query_dhcp_reg_info: DHCP
provisioned option 125 available
Nov 18 17:21:00.965: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: parsing
data in option 125 of length 73
Nov 18 17:21:00.965: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: enterprise
 ID 210
Nov 18 17:21:00.965: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: total option
 data length 80
Nov 18 17:21:00.965: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: sub-option
 type 201 of length 6
Nov 18 17:21:00.965: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_macaddr: MAC
 addr 1234567890AB
Nov 18 17:21:00.969:
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: sub-option
 type 202 of length 6
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_contract_num:
 pilot # 777777
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: sub-option
 type 203 of length 6
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_addn_num:
secondary # 222222 (index 0)
```

```
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: sub-option
 type 203 of length 6
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_addn_num:
secondary # 333333 (index 1)
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: sub-option
 type 203 of length 6
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_addn_num:
secondary # 444444 (index 2)
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: sub-option
 type 203 of length 6
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_addn_num:
secondary # 555555 (index 3)
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: sub-option
 type 203 of length 6
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_addn_num:
secondary # 666666 (index 4)
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: sub-option
 type 204 of length 14
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_domain:
domain sublen 5
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_domain:
domain sublen 3
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_subopt_domain:
domain dns:cisco.com
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/ccsip_gw_parse_dhcp_opt125: parsing of
 DHCP option 125 succeeded
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/SIP-DHCP/sipua_query_dhcp_reg_info: DHCP
provisioned SIP server addr: 9.13.2.36
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_register_cred_user: Sending msg type
 2 to register process from parser for user 777777
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_spi_register_process_e164_registration:
 CCSIP_REGISTER:: e164 number (777777)
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_register_search_e164_table: ****No
entry found in E164 Table
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Info/sipSPIAddContextToTable: Added
context(0x476FD758) with key=[1061] to table
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Info/sipSPIGetOutboundHostAndDestHostPrivate:
CCSIP: target_host : cisco.com target_port : 5060
Nov 18 17:21:00.969: //-1/000000000000/SIP/Info/sipSPIValidateAndCopyOutboundHost: CCSIP:
copy target_host to outbound_host
Nov 18 17:21:00.969: //-1/000000000000/SIP/State/sipSPIChangeState: 0x476FD758 : State
change from (STATE_NONE, SUBSTATE_NONE)  to (STATE_IDLE, SUBSTATE_NONE)
Nov 18 17:21:00.969: //-1/000000000000/SIP/Info/ccsip_spi_registrar_add_expires_header:
Inside ccsip_spi_registrar_add_expires_header for Expires
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Event/sipSPIEventInfo: Queued event from SIP SPI
 : SIPSPI_EV_OUTBOUND_REGISTER
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_register_add_to_e164_table: ****Added
 to E164 Table
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_process_sipspi_queue_event:
ccsip_spi_get_msg_type returned: 3 for event 40
Nov 18 17:21:00.969: //-1/000000000000/SIP/Info/act_idle_outgoing_register: In
act_idle_outgoing_register
Nov 18 17:21:00.969: //1034/000000000000/SIP/Info/act_idle_outgoing_register:  Send REGISTER
 to cisco.com:5060
Nov 18 17:21:00.969: //1034/000000000000/SIP/Info/sipSPIUaddCcbToUACTable: ****Adding to
UAC table.
Nov 18 17:21:00.969: //1034/000000000000/SIP/Info/sipSPIUaddCcbToTable: Added to table.
ccb=0x476FD758 key=1AF6E28A-B4CC11DD-81078B9C-6E99E02B
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Event/sipSPIEventInfo: Queued event from SIP SPI
 : SIPSPI_EV_DNS_RESOLVE
Nov 18 17:21:00.969: //1034/000000000000/SIP/State/sipSPIChangeState: 0x476FD758 : State
change from (STATE_IDLE, SUBSTATE_NONE)  to (STATE_IDLE, SUBSTATE_SENT_DNS)
Nov 18 17:21:00.969: //1034/000000000000/SIP/State/sipSPIChangeState: 0x476FD758 : State
change from (STATE_IDLE, SUBSTATE_SENT_DNS)  to (SIP_STATE_OUTGOING_REGISTER,
SUBSTATE_SENT_DNS)
Nov 18 17:21:00.969: //-1/xxxxxxxxxxxx/SIP/Info/sip_dns_type_srv_query: TYPE SRV query for
 _sip._udp.cisco.com and type:1
Nov 18 17:21:00.977: //-1/xxxxxxxxxxxx/SIP/Info/sip_dns_type_a_aaaa_query: DNS query for
cisco.com and type:1
Nov 18 17:21:00.977: //-1/xxxxxxxxxxxx/SIP/Info/sip_dns_type_a_query: TYPE A query successful
 for cisco.com
Nov 18 17:21:00.977: //-1/xxxxxxxxxxxx/SIP/Info/sip_dns_type_a_aaaa_query: IP Address of
cisco.com is:
```

```
Nov 18 17:21:00.977: //-1/xxxxxxxxxxxx/SIP/Info/sip_dns_type_a_aaaa_query: 9.13.2.36
Nov 18 17:21:00.977: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_process_sipspi_queue_event:
ccsip_spi_get_msg_type returned: 2 for event 43
Nov 18 17:21:00.977: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_register_set_dns_resolved_address:
CCSIP_REGISTER:: Primary registrar DNS resolved addr set to 0.0.0.1:151847460
Nov 18 17:21:00.977: //-1/xxxxxxxxxxxx/SIP/Info/ccsipRegisterStartExpiresTimer: Starting
timer for pattern  for 3600 seconds
Nov 18 17:21:00.977: //1034/000000000000/SIP/State/sipSPIChangeState: 0x476FD758 : State
change from (SIP_STATE_OUTGOING_REGISTER, SUBSTATE_SENT_DNS)  to (SIP_STATE_OUTGOING_REGISTER,
 SUBSTATE_NONE)
Nov 18 17:21:00.977: //-1/xxxxxxxxxxxx/SIP/Info/sipSPISetDateHeader: Clock Time Zone is
UTC, same as GMT: Using GMT
Nov 18 17:21:00.981: //1034/000000000000/SIP/Info/sipSPISendRegister: Associated
container=0x46794ACC to Register
Nov 18 17:21:00.981: //1034/000000000000/SIP/Transport/sipSPISendRegister: Sending REGISTER
 to the transport layer
Nov 18 17:21:00.981: //1034/000000000000/SIP/Transport/sipSPIGetSwitchTransportFlag: Return
 the Dial peer configuration, Switch Transport is FALSE
Nov 18 17:21:00.981: //1034/000000000000/SIP/Transport/sipSPITransportSendMessage:
msg=0x4707F998, addr=9.13.2.36, port=5060, sentBy_port=0, is_req=1, transport=1, switch=0,
 callBack=0x415A53B0
Nov 18 17:21:00.981: //1034/000000000000/SIP/Transport/sipSPITransportSendMessage: Proceedable
 for sending msg immediately
Nov 18 17:21:00.981: //1034/000000000000/SIP/Transport/sipTransportLogicSendMsg: switch
transport is 0
Nov 18 17:21:00.981: //1034/000000000000/SIP/Transport/sipTransportLogicSendMsg: Set to
send the msg=0x4707F998
Nov 18 17:21:00.981: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportPostSendMessage: Posting
send for msg=0x4707F998, addr=9.13.2.36, port=5060, connId=2 for UDP
Nov 18 17:21:00.981: //1034/000000000000/SIP/State/sipSPIChangeState: 0x476FD758 : State
change from (SIP_STATE_OUTGOING_REGISTER, SUBSTATE_NONE)  to (SIP_STATE_OUTGOING_REGISTER,
 SUBSTATE_NONE)
Nov 18 17:21:00.981: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
REGISTER sip:cisco.com:5060 SIP/2.0
Date: Tue, 18 Nov 2008 17:21:00 GMT
From: <sip:777777@cisco.com>;tag=34FBAED8-131
Supported: path
Timestamp: 1227028860
Content-Length: 0
User-Agent: Cisco-SIPGateway/IOS-12.x
To: <sip:777777@cisco.com>
Contact: <sip:777777@9.13.8.183:5060>
Expires:  3600
Call-ID: 1AF6E28A-B4CC11DD-81078B9C-6E99E02B
Via: SIP/2.0/UDP 9.13.8.183:5060;branch=z9hG4bK3F522D9
CSeq: 2 REGISTER
Max-Forwards: 70
Nov 18 17:21:00.981: //-1/xxxxxxxxxxxx/SIP/Info/HandleUdpIPv4SocketReads: Msg enqueued for
 SPI with IP addr: [9.13.2.36]:56305
Nov 18 17:21:00.981: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_process_sipspi_queue_event:
ccsip_spi_get_msg_type returned: 2 for event 1
Nov 18 17:21:00.981: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportProcessNWNewConnMsg:
context=0x00000000
Nov 18 17:21:00.985: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_new_msg_preprocessor: Checking Invite
 Dialog
Nov 18 17:21:00.985: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 9.13.8.183:5060;received=9.13.8.183;branch=z9hG4bK3F522D9
Call-ID: 1AF6E28A-B4CC11DD-81078B9C-6E99E02B
From: <sip:777777@cisco.com>;tag=34FBAED8-131
To: <sip:777777@cisco.com>
CSeq: 2 REGISTER
Content-Length: 0
Nov 18 17:21:01.077: //-1/xxxxxxxxxxxx/SIP/Info/HandleUdpIPv4SocketReads: Msg enqueued for
 SPI with IP addr: [9.13.2.36]:56306
Nov 18 17:21:01.077: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_process_sipspi_queue_event:
ccsip_spi_get_msg_type returned: 2 for event 1
Nov 18 17:21:01.077: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportProcessNWNewConnMsg:
context=0x00000000
Nov 18 17:21:01.077: //-1/xxxxxxxxxxxx/SIP/Info/ccsip_new_msg_preprocessor: Checking Invite
 Dialog
```

```
Nov 18 17:21:01.077: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 9.13.8.183:5060;received=9.13.8.183;branch=z9hG4bK3F522D9
Call-ID: 1AF6E28A-B4CC11DD-81078B9C-6E99E02B
From: <sip:777777@cisco.com>;tag=34FBAED8-131
To: <sip:777777@cisco.com>
CSeq: 2 REGISTER
Contact: <sip:777777@9.13.8.183:5060>;expires=3600
Content-Length: 0
Nov 18 17:21:01.077: //1034/000000000000/SIP/Info/ccsip_gw_register_process_response: No
P-Assoicated-URI present in Register Response
Nov 18 17:21:01.077: //-1/xxxxxxxxxxxx/SIP/Info/ccsipRegisterStartExpiresTimer: Starting
timer for pattern 777777 for 2880 seconds
Nov 18 17:21:01.077: //-1/xxxxxxxxxxxx/SIP/Info/sipSPIDeleteContextFromTable: Context for
key=[1061] removed.
Nov 18 17:21:01.077: //1034/000000000000/SIP/Info/sipSPIUdeleteCcbFromUACTable: ****Deleting
 from UAC table.
Nov 18 17:21:01.077: //1034/000000000000/SIP/Info/sipSPIUdeleteCcbFromTable: Deleting from
 table. ccb=0x476FD758 key=1AF6E28A-B4CC11DD-81078B9C-6E99E02B
Nov 18 17:21:01.077: //1034/000000000000/SIP/Info/sipSPIFlushEventBufferQueue: There are 0
 events on the internal queue that are going to be free'd
Nov 18 17:21:01.077: //1034/000000000000/SIP/Info/ccsip_qos_cleanup: Entry
Nov 18 17:21:01.077: //1034/000000000000/SIP/Info/sipSPI_ipip_free_codec_profile: Codec
Profiles Freed
Nov 18 17:21:01.077: //1034/000000000000/SIP/Info/sipSPIUfreeOneCCB: Freeing ccb 476FD758
Nov 18 17:21:01.081: //-1/xxxxxxxxxxxx/SIP/Info/sipSPIGetContextFromTable: NO context for
key[1061]
Nov 18 17:21:02.761: %SYS-5-CONFIG_I: Configured from console by console
CUBE-DHCP-CLIENT1#
```

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging
debug ccsip calls	Displays all SIP SPI call tracing.
debug ccsip error	Displays SIP SPI errors.
debug ccsip events	Displays all SIP SPI events tracing.
debug ccsip in	Displays all SIP SPI message tracing.
debug ccsip states	Displays all SIP SPI state tracing.

debug ccsip error

To show Session Initiation Protocol (SIP) Service Provider Interface (SPI) errors, use the **debugccsiperror** command in privileged EXEC mode. To disable debugging output, use the no form of this command.

debug ccsip error

no debug ccip error

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

12.1(1)T	This command was introduced.
12.2(2)XA	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB1	This command was implemented on the Cisco AS5850 universal gateway.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on Cisco 7200 series routers.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T. Support for the Cisco AS5300 universal access server, Cisco AS5350, Cisco AS5400, and Cisco AS5850 universal gateway is not included in this release.

Usage Guidelines

This command traces all error messages generated from errors encountered by the SIP subsystem.

Examples

The following example displays debug output from one side of the call:

```
Router1#
debug ccsip error
SIP Call error tracing is enabled
Router1#
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  act_idle_call_setup
*Mar  6 14:16:41:  act_idle_call_setup:Not using Voice Class Codec
*Mar  6 14:16:41: act_idle_call_setup: preferred_codec set[0] type :g711ulaw bytes: 160
*Mar  6 14:16:41: REQUEST CONNECTION TO IP:166.34.245.231 PORT:5060
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  act_idle_connection_created
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  act_idle_connection_created: Connid(1) created to
166.34.245.231:5060, local_port 55674
*Mar  6 14:16:41: sipSPIAddLocalContact
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  6 14:16:41: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.231:5060
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  act_sentinvite_new_message
```

```
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:16:41:  Roundtrip delay 4 milliseconds for method INVITE
*Mar  6 14:16:41: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.231:5060
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  act_recdproc_new_message
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  sipSPICheckResponse : Updating session description
*Mar  6 14:16:41: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:16:41:  Roundtrip delay 8 milliseconds for method INVITE
*Mar  6 14:16:41: HandleSIP1xxRinging: SDP MediaTypes negotiation successful!
Negotiated Codec      : g711ulaw , bytes :160
Inband Alerting       : 0
*Mar  6 14:16:45: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.231:5060
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  act_recdproc_new_message
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  sipSPICheckResponse : Updating session description
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:16:45:  Roundtrip delay 3844 milliseconds for method INVITE
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  act_recdproc_new_message: SDP MediaTypes negotiation
 successful!
Negotiated Codec      : g711ulaw , bytes :160
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  sipSPIReconnectConnection
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  recv_200_OK_for_invite
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  6 14:16:45: HandleUdpReconnection: Udp socket connected for fd: 1 with
166.34.245.231:5060
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  ccsip_caps_ind
*Mar  6 14:16:45: ccsip_caps_ind: Load DSP with codec (5) g711ulaw, Bytes=160
*Mar  6 14:16:45: ccsip_caps_ind: set DSP for dtmf-relay = CC_CAP_DTMF_RELAY_INBAND_VOICE
*Mar  6 14:16:45: CCSIP-SPI-CONTROL:  ccsip_caps_ack
*Mar  6 14:16:49: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.231:56101
*Mar  6 14:16:49: CCSIP-SPI-CONTROL:  act_active_new_message
*Mar  6 14:16:49: CCSIP-SPI-CONTROL:  sact_active_new_message_request
*Mar  6 14:16:49: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  6 14:16:49: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:16:49: CCSIP-SPI-CONTROL:  sipSPIInitiateCallDisconnect : Initiate call
disconnect(16) for outgoing call
*Mar  6 14:16:49: CCSIP-SPI-CONTROL:  act_disconnecting_disconnect
*Mar  6 14:16:49: CCSIP-SPI-CONTROL:  sipSPICallCleanup
*Mar  6 14:16:49: CLOSE CONNECTION TO CONNID:1
*Mar  6 14:16:49: sipSPIIcpifUpdate :CallState: 4 Playout: 2945 DiscTime:48340988 ConnTime
 48340525
*Mar  6 14:16:49: udpsock_close_connect: Socket fd: 1 closed for connid 1 with remote port:
 5060
```

The following example displays debug output from the other side of the call:

```
Router2# debug ccsip error
SIP Call error tracing is enabled
Router2#
*Mar  8 17:42:39: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.230:55674
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  sipSPISipIncomingCall
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  act_idle_new_message
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  sact_idle_new_message_invite
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:42:39:  sact_idle_new_message_invite:Not Using Voice Class Codec
*Mar  8 17:42:39: sact_idle_new_message_invite: Preferred codec[0] type: g711ulaw Bytes
:160
*Mar  8 17:42:39: sact_idle_new_message_invite: Media Negotiation successful for an
incoming call
*Mar  8 17:42:39: sact_idle_new_message_invite: Negotiated Codec      : g711ulaw, bytes
:160
Preferred Codec       : g711ulaw, bytes :160
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:42:39: Num of Contact Locations 1 3660110 166.34.245.230 5060
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  act_recdinvite_proceeding
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  ccsip_caps_ind
*Mar  8 17:42:39: ccsip_caps_ind: codec(negotiated) = 5(Bytes 160)
*Mar  8 17:42:39: ccsip_caps_ind: Load DSP with codec (5) g711ulaw, Bytes=160
*Mar  8 17:42:39: ccsip_caps_ind: set DSP for dtmf-relay = CC_CAP_DTMF_RELAY_INBAND_VOICE
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  ccsip_caps_ack
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  act_recdinvite_alerting
*Mar  8 17:42:39:  180 Ringing with SDP - not likely
```

```
*Mar  8 17:42:39: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:42:42: CCSIP-SPI-CONTROL:  act_sentalert_connect
*Mar  8 17:42:42: sipSPIAddLocalContact
*Mar  8 17:42:42: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:42:42: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.230:55674
*Mar  8 17:42:42: CCSIP-SPI-CONTROL:  act_sentsucc_new_message
*Mar  8 17:42:42: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  act_active_disconnect
*Mar  8 17:42:47: REQUEST CONNECTION TO IP:166.34.245.230 PORT:5060
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  act_active_connection_created
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  sipSPICheckSocketConnection
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  sipSPICheckSocketConnection: Connid(1) created to
166.34.245.230:5060, local_port 56101
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:42:47: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 166.34.245.230:55674
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  act_disconnecting_new_message
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  sact_disconnecting_new_message_response
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:42:47:  Roundtrip delay 0 milliseconds for method BYE
*Mar  8 17:42:47: CCSIP-SPI-CONTROL:  sipSPICallCleanup
*Mar  8 17:42:47: CLOSE CONNECTION TO CONNID:1
*Mar  8 17:42:47: sipSPIIcpifUpdate :CallState: 4 Playout: 1255 DiscTime:66856757 ConnTime
 66856294
*Mar  8 17:42:47: udpsock_close_connect: Socket fd: 1 closed for connid 1 with remote port:
 5060
```

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging.
debug ccsip calls	Shows all SIP SPI call tracing.
debug ccsip events	Shows all SIP SPI events tracing.
debug ccsip info	Shows all SIP SPI message tracing.
debug ccsip states	Shows all SIP SPI state tracing.

debug ccsip events

To enable tracing of events that are specific to service provider interface (SPI), use the **debugccsipevents**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccsip events

no debug ccsip events

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

12.1(1)T	This command was introduced.
12.2(2)XA	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB1	This command was introduced on the Cisco AS5850 universal gateway.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T.
12.2(15)T	Much of the information formerly found in the output of the **debugccsipevents** command is now reported in the output of the **debugccsipinfo** and **debugccsipmedia**commands. The **debugccsipevents**command now displays only the debugging information specifically related to SIP events.

Usage Guidelines

This command previously traced all events posted to Session Initiation Protocol (SIP) SPI from all interfaces and also provided general SIP SPI information. Beginning with Cisco IOS Release 12.2(15)T, the **debugccsipevents** command displays only debugging information specifically related to SIP SPI events. Media stream and SIP SPI information is now reported in the **debugccsipmedia**and **debugccsipinfo** command output.

Note This command is intended for use by Cisco technicians only.

Examples

The following is sample output from the **debugccsipevents**command for a Cisco 3660:

```
Router# debug ccsip events
SIP Call events tracing is enabled
```

```
Router#
Nov 15 18:20:25.779: Queued event from SIP SPI : SIPSPI_EV_CC_CALL_SETUP
Nov 15 18:20:25.779: Queued event from SIP SPI : SIPSPI_EV_CREATE_CONNECTION
Nov 15 18:20:25.783: Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
Nov 15 18:20:25.815: Queued event from SIP SPI : SIPSPI_EV_CREATE_CONNECTION
Nov 15 18:20:25.819: Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
Nov 15 18:20:28.339: Queued event from SIP SPI : SIPSPI_EV_CLOSE_CONNECTION
Nov 15 18:20:28.339: Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
Nov 15 18:20:50.844: Queued event from SIP SPI : SIPSPI_EV_CLOSE_CONNECTION
Nov 15 18:20:50.844: Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
Nov 15 18:20:50.848: Queued event from SIP SPI : SIPSPI_EV_CC_CALL_DISCONNECT
```

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging.
debug ccsip info	Enables tracing of general SIP SPI information.
debug ccsip media	Enables tracing of SIP call media streams.

debug ccsip info

To enable tracing of general service provider interface (SPI) information, use the **debugccsipinfo**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccsip info

no debug ccsip info

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.

Usage Guidelines

Beginning in Cisco IOS Release 12.2(15)T, the **debugccsipinfo**command is a separate option that displays general SIP SPI information for debug purposes. In past releases, this output was part of the **debugccsipevents**command.

Note This command is intended for use by Cisco technicians only.

Examples

The following is sample output from the **debugccsipinfo**command for a Cisco 3660:

```
Router# debug ccsip info
SIP Call info tracing is enabled
Router#
Nov 15 18:19:22.670: ****Adding to UAC table
Nov 15 18:19:22.670: adding call id E to table
Nov 15 18:19:22.670: CCSIP-SPI-CONTROL: act_idle_call_setup
Nov 15 18:19:22.670: act_idle_call_setup:Not using Voice Class Codec
Nov 15 18:19:22.670: act_idle_call_setup: preferred_codec set[0] type :g729r8 bytes: 20
Nov 15 18:19:22.670: sipSPICopyPeerDataToCCB: From CLI: Modem NSE payload = 100, Passthrough
 = 0,Modem relay = 0, Gw-Xid = 1
SPRT latency 200, SPRT Retries = 12, Dict Size = 1024
String Len = 32, Compress dir = 3
Nov 15 18:19:22.670: ****Deleting from UAC table
Nov 15 18:19:22.670: ****Adding to UAC table
Nov 15 18:19:22.670: sipSPIUsetBillingProfile: sipCallId for billing records =
20A40C3B-D92C11D5-8015E1CC-C91F3F10@12.18.195.49
Nov 15 18:19:22.674: CCSIP-SPI-CONTROL: act_idle_connection_created
Nov 15 18:19:22.674: CCSIP-SPI-CONTROL: act_idle_connection_created: Connid(1) created to
172.18.193.190:5060, local_port 56981
Nov 15 18:19:22.674: CCSIP-SPI-CONTROL: sipSPIOutgoingCallSDP
Nov 15 18:19:22.674: convert_codec_bytes_to_ptime: Values :Codec: g729r8 codecbytes :20,
ptime: 10
Nov 15 18:19:22.674: sip_generate_sdp_xcaps_list: Modem Relay disabled. X-cap not needed
```

```
Nov 15 18:19:22.674: sipSPIAddLocalContact
Nov 15 18:19:22.674: sip_stats_method
Nov 15 18:19:22.690: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.190:5060
Nov 15 18:19:22.690: CCSIP-SPI-CONTROL: act_sentinvite_new_message
Nov 15 18:19:22.690: CCSIP-SPI-CONTROL: sipSPICheckResponse
Nov 15 18:19:22.690: sip_stats_status_code
Nov 15 18:19:22.690: Roundtrip delay 16 milliseconds for method INVITE
Nov 15 18:19:22.706: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.190:5060
Nov 15 18:19:22.706: CCSIP-SPI-CONTROL: act_recdproc_new_message
Nov 15 18:19:22.706: CCSIP-SPI-CONTROL: sipSPICheckResponse
Nov 15 18:19:22.706: sip_stats_status_code
Nov 15 18:19:22.706: Roundtrip delay 32 milliseconds for method INVITE
Nov 15 18:19:22.706: sipSPIGetSdpBody : Parse incoming session description
Nov 15 18:19:22.706: HandleSIP1xxSessionProgress: Content-Disposition received in 18x
response:session;handling=required
Nov 15 18:19:22.706: sipSPIDoMediaNegotiation: number of m lines is 1
Nov 15 18:19:22.706: sipSPIDoAudioNegotiation: Codec (g729r8) Negotiation Successful on
Static Payload
Nov 15 18:19:22.706: sipSPIDoPtimeNegotiation: One ptime attribute found - value:10
Nov 15 18:19:22.706: convert_ptime_to_codec_bytes: Values :Codec: g729r8 ptime :10,
codecbytes: 20
Nov 15 18:19:22.710: convert_codec_bytes_to_ptime: Values :Codec: g729r8 codecbytes :20,
ptime: 10
Nov 15 18:19:22.710: sipSPIDoDTMFRelayNegotiation: m-line index 1
Nov 15 18:19:22.710: sipSPIDoDTMFRelayNegotiation: Requested DTMF-RELAY option(s) not found
 in Preferred DTMF-RELAY option list!
Nov 15 18:19:22.710: sip_sdp_get_modem_relay_cap_params:
Nov 15 18:19:22.710: sip_sdp_get_modem_relay_cap_params: NSE payload from X-cap = 0
Nov 15 18:19:22.710: sip_do_nse_negotiation: NSE Payload 100 found in SDP
Nov 15 18:19:22.710: sip_do_nse_negotiation: Remote NSE payload = local one = 100, Use it
Nov 15 18:19:22.710: sip_select_modem_relay_params: X-tmr not present in SDP. Disable modem
 relay
Nov 15 18:19:22.710: sipSPIDoQoSNegotiation - SDP body with media description
Nov 15 18:19:22.710: ccsip_process_response_contact_record_route
Nov 15 18:19:22.710: CCSIP-SPI-CONTROL: ccsip_bridge: confID = 4, srcCallID = 14, dstCallID
 = 13
Nov 15 18:19:22.710: sipSPIUupdateCcCallIds: old src/dest ccCallids: -1/-1, new src/dest
ccCallids: 14/13
Nov 15 18:19:22.710: sipSPIUupdateCcCallIds: old streamcallid=-1, new streamcallid=14
Nov 15 18:19:22.710: CCSIP-SPI-CONTROL: ccsip_caps_ind
Nov 15 18:19:22.710: ccsip_get_rtcp_session_parameters: CURRENT VALUES: stream_callid=14,
current_seq_num=0x1B1B
Nov 15 18:19:22.710: ccsip_get_rtcp_session_parameters: NEW VALUES: stream_callid=14,
current_seq_num=0x180C
Nov 15 18:19:22.710: ccsip_caps_ind: Load DSP with negotiated codec : g729r8, Bytes=20
Nov 15 18:19:22.710: ccsip_caps_ind: set forking flag to 0x0
Nov 15 18:19:22.710: sipSPISetDTMFRelayMode: set DSP for dtmf-relay =
CC_CAP_DTMF_RELAY_INBAND_VOICE_AND_OOB
Nov 15 18:19:22.710: sip_set_modem_caps: Negotiation already Done. Set negotiated Modem
caps
Nov 15 18:19:22.710: sip_set_modem_caps: Modem Relay & Passthru both disabled
Nov 15 18:19:22.710: sip_set_modem_caps: nse payload = 100, ptru mode = 0, ptru-codec=0,
redundancy=0, xid=0, relay=0, sprt-retry=12, latecncy=200, compres-dir=3, dict=1024,
strnlen=32
Nov 15 18:19:22.710: ccsip_caps_ind: Load DSP with codec : g729r8, Bytes=20
Nov 15 18:19:22.710: CCSIP-SPI-CONTROL: ccsip_caps_ack
Nov 15 18:19:22.710: ccsip_caps_ack: set forking flag to 0x60FD1EAC
Nov 15 18:19:22.710: CCSIP-SPI-CONTROL: act_recdproc_connection_created
Nov 15 18:19:22.710: CCSIP-SPI-CONTROL: sipSPICheckSocketConnection: Connid(2) created to
172.18.193.190:5060, local_port 51663
Nov 15 18:19:22.714: sip_stats_method
Nov 15 18:19:22.722: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.190:5060
Nov 15 18:19:22.722: CCSIP-SPI-CONTROL: act_recdproc_new_message
Nov 15 18:19:22.722: CCSIP-SPI-CONTROL: sipSPICheckResponse
Nov 15 18:19:22.722: sip_stats_status_code
Nov 15 18:19:22.722: Roundtrip delay 48 milliseconds for method PRACK
Nov 15 18:19:24.706: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.190:5060
Nov 15 18:19:24.706: CCSIP-SPI-CONTROL: act_recdproc_new_message
Nov 15 18:19:24.706: CCSIP-SPI-CONTROL: sipSPICheckResponse
```

```
Nov 15 18:19:24.706: sip_stats_status_code
Nov 15 18:19:24.706: Roundtrip delay 2032 milliseconds for method PRACK
Nov 15 18:19:24.706: sipSPIGetSdpBody : Parse incoming session description
Nov 15 18:19:24.710: CCSIP-SPI-CONTROL: sipSPIUACSessionTimer
Nov 15 18:19:24.710: CCSIP-SPI-CONTROL: act_recdproc_continue_200_processing
Nov 15 18:19:24.710: CCSIP-SPI-CONTROL: act_recdproc_continue_200_processing: *** This ccb
 is the parent
Nov 15 18:19:24.710: sipSPICompareRespMediaInfo
Nov 15 18:19:24.710: sipSPIDoMediaNegotiation: number of m lines is 1
Nov 15 18:19:24.710: sipSPIDoAudioNegotiation: Codec (g729r8) Negotiation Successful on
Static Payload
Nov 15 18:19:24.710: sipSPIDoPtimeNegotiation: One ptime attribute found - value:10
Nov 15 18:19:24.710: convert_ptime_to_codec_bytes: Values :Codec: g729r8 ptime :10,
codecbytes: 20
Nov 15 18:19:24.710: convert_codec_bytes_to_ptime: Values :Codec: g729r8 codecbytes :20,
ptime: 10
Nov 15 18:19:24.710: sipSPIDoDTMFRelayNegotiation: m-line index 1
Nov 15 18:19:24.710: sipSPIDoDTMFRelayNegotiation: Requested DTMF-RELAY option(s) not found
 in Preferred DTMF-RELAY option list!
Nov 15 18:19:24.710: sip_sdp_get_modem_relay_cap_params:
Nov 15 18:19:24.710: sip_sdp_get_modem_relay_cap_params: NSE payload from X-cap = 0
Nov 15 18:19:24.710: sip_do_nse_negotiation: NSE Payload 100 found in SDP
Nov 15 18:19:24.710: sip_do_nse_negotiation: Remote NSE payload = local one = 100, Use it
Nov 15 18:19:24.710: sip_select_modem_relay_params: X-tmr not present in SDP. Disable modem
 relay
Nov 15 18:19:24.710: sipSPIProcessMediaChanges
Nov 15 18:19:24.710: ccsip_process_response_contact_record_route
Nov 15 18:19:24.710: CCSIP-SPI-CONTROL: sipSPIProcess200OKforinvite
Nov 15 18:19:24.710: sip_stats_method
Nov 15 18:19:24.710: udpsock_close_connect: Socket fd: 1 closed for connid 1 with remote
port: 5060
Nov 15 18:19:37.479: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.190:52180
Nov 15 18:19:37.483: ****Found CCB in UAC table
Nov 15 18:19:37.483: CCSIP-SPI-CONTROL: act_active_new_message
Nov 15 18:19:37.483: CCSIP-SPI-CONTROL: sact_active_new_message_request
Nov 15 18:19:37.483: sip_stats_method
Nov 15 18:19:37.483: sip_stats_status_code
Nov 15 18:19:37.483: CCSIP-SPI-CONTROL: sipSPIInitiateCallDisconnect : Initiate call
disconnect(16) for outgoing call
Nov 15 18:19:37.483: udpsock_close_connect: Socket fd: 2 closed for connid 2 with remote
port: 5060
Nov 15 18:19:37.483: CCSIP-SPI-CONTROL: act_disconnecting_disconnect
Nov 15 18:19:37.483: CCSIP-SPI-CONTROL: sipSPICallCleanup
Nov 15 18:19:37.483: sipSPIIcpifUpdate :CallState: 4 Playout: 10230 DiscTime:1745148 ConnTime
 1743871
Nov 15 18:19:37.483: ****Deleting from UAC table
Nov 15 18:19:37.483: Removing call id E
Nov 15 18:19:37.483: freeing ccb 63330954
```

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging.
debug ccsip events	Enables tracing of events that are specific to SIP SPI.
debug ccsip media	Enables tracing of SIP call media streams.

debug ccsip media

To enable tracing of Session Initiation Protocol (SIP) call media streams, use the **debugccsipmedia**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccsip media

no debug ccsip media

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.

Usage Guidelines

Beginning in Cisco IOS Release 12.2(15)T, the **debugccsipmedia**command is a separate option that displays debugging information specific to SIP media stream processing. In past releases, this output was part of the **debugccsipevents**command.

Note This command is intended for use by Cisco technicians only.

Examples

The following is sample output from the **debugccsipmedia**command for a Cisco 3660:

```
Router# debug ccsip media

SIP Call media tracing is enabled
Router#
Nov 15 18:19:53.835: sipSPISetMediaSrcAddr: media src addr for stream 1 = 172.18.195.49
Nov 15 18:19:53.835: sipSPIReserveRtpPort: reserved port 16500 for stream 1
Nov 15 18:19:53.867: sipSPIReplaceSDP
Nov 15 18:19:53.871: sipSPICopySdpInfo
Nov 15 18:19:53.871: sipSPIUpdCallWithSdpInfo:
Preferred Codec : g729r8, bytes :20
Preferred DTMF relay : inband-voice
Preferred NTE payload : 101
Early Media : No
Delayed Media : No
Bridge Done : No
New Media : No
DSP DNLD Reqd : No
Nov 15 18:19:53.871: sipSPISetMediaSrcAddr: media src addr for stream 1 = 172.18.195.49
Nov 15 18:19:53.871: sipSPIUpdCallWithSdpInfo:
M-line Index : 1
State : STREAM_ADDING (3)
Callid : -1
Negotiated Codec : g729r8, bytes :20
```

```
Negotiated DTMF relay : inband-voice
Negotiated NTE payload : 0
Media Srce Addr/Port : 172.18.195.49:16500
Media Dest Addr/Port : 172.18.193.190:19148
Nov 15 18:19:53.871: sipSPIProcessRtpSessions
Nov 15 18:19:53.871: sipSPIAddStream: Adding stream 1 (callid 16) to the VOIP RTP library
Nov 15 18:19:53.871: sipSPISetMediaSrcAddr: media src addr for stream 1 = 172.18.195.49
Nov 15 18:19:53.871: sipSPIUpdateRtcpSession: for m-line 1
Nov 15 18:19:53.871: sipSPIUpdateRtcpSession: rtcp_session info
laddr = 172.18.195.49, lport = 16500, raddr = 172.18.193.190, rport=19148
Nov 15 18:19:53.871: sipSPIUpdateRtcpSession: No rtp session, creating a new one
Nov 15 18:19:53.871: sipSPISetStreamInfo: num_streams = 1
Nov 15 18:19:53.871: sipSPISetStreamInfo: adding stream type 0 from mline 1
Nov 15 18:19:53.871: sipSPISetStreamInfo: caps.stream_count=1, caps.stream[0].stream_type=0x1,
 caps.stream_list.xmitFunc=voip_rtp_xmit, caps.stream_list.context=0x634F1F2C (gccb)
Nov 15 18:19:55.555: sipSPICompareSDP
Nov 15 18:19:55.555: sipSPICompareStreams: stream 1 dest_port: old=19148 new=19148
Nov 15 18:19:55.555: sipSPICompareStreams: Flags set for stream 1: RTP_CHANGE=No
CAPS_CHANGE=No
Nov 15 18:19:55.555: sipSPICompareSDP: Flags set for call: NEW_MEDIA=No DSPDNLD_REQD=No
Nov 15 18:19:55.555: sipSPIReplaceSDP
Nov 15 18:19:55.555: sipSPICopySdpInfo
Nov 15 18:19:55.555: sipSPIUpdCallWithSdpInfo:
Preferred Codec : g729r8, bytes :20
Preferred DTMF relay : inband-voice
Preferred NTE payload : 101
Early Media : No
Delayed Media : No
Bridge Done : Yes
New Media : No
DSP DNLD Reqd : No
Nov 15 18:19:55.555: sipSPISetMediaSrcAddr: media src addr for stream 1 = 172.18.195.49
Nov 15 18:19:55.555: sipSPIUpdCallWithSdpInfo:
M-line Index : 1
State : STREAM_ACTIVE (3)
Callid : 16
Negotiated Codec : g729r8, bytes :20
Negotiated DTMF relay : inband-voice
Negotiated NTE payload : 0
Media Srce Addr/Port : 172.18.195.49:16500
Media Dest Addr/Port : 172.18.193.190:19148
```

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging.
debug ccsip events	Enables tracing of events that are specific to SIP SPI.
debug ccsip info	Enables tracing of general SIP SPI events.

debug ccsip messages

To show all Session Initiation Protocol (SIP) Service Provider Interface (SPI) message tracing, use the **debugccsipmessages** command in privileged EXEC mode. To disable debugging output, use the no form of this command.

debug ccsip messages

no debug ccsip messages

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.1.(3)T	The output of this command was changed.
12.2(2)XA	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB1	This command was implemented on the Cisco AS5850 universal gateway.
12.2(8)T	This command was implemented on Cisco 7200 series routers.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T. Support for the Cisco AS5300 universal access server, Cisco AS5350, Cisco AS5400, and Cisco AS5850 universal gateway is not included in this release.
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.
IOS Release XE 2.5	This command was integrated into Cisco IOS XE Release 2.5.

Usage Guidelines

This command traces the Session Initiation Protocol (SIP) messages exchanged between the SIP UA client (UAC) and the access server.

Examples

The following example shows debug output from one side of the call:

```
Router1#
debug ccsip messages
SIP Call messages tracing is enabled
Router1#
*Mar  6 14:19:14: Sent:
INVITE sip:3660210@166.34.245.231;user=phone;phone-context=unknown SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Sat, 06 Mar 1993 19:19:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Cisco-Guid: 2881152943-2184249568-0-483551624
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731427554
Contact: <sip:3660110@166.34.245.230:5060;user=phone>
Expires: 180
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 5596 7982 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20762 RTP/AVP 0
*Mar  6 14:19:14: Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Length: 0
*Mar  6 14:19:14: Received:
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 1193 7927 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20224 RTP/AVP 0
*Mar  6 14:19:16: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Contact: <sip:3660210@166.34.245.231:5060;user=phone>
CSeq: 101 INVITE
Content-Type: application/sdp
```

```
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 1193 7927 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20224 RTP/AVP 0
*Mar  6 14:19:16: Sent:
ACK sip:3660210@166.34.245.231:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
Date: Sat, 06 Mar 1993 19:19:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Max-Forwards: 6
Content-Type: application/sdp
Content-Length: 138
CSeq: 101 ACK
v=0
o=CiscoSystemsSIP-GW-UserAgent 5596 7982 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20762 RTP/AVP 0
*Mar  6 14:19:19: Received:
BYE sip:3660110@166.34.245.230:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.231:53600
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
To: "3660110" <sip:3660110@166.34.245.230>
Date: Mon, 08 Mar 1993 22:45:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 6
Timestamp: 731612717
CSeq: 101 BYE
Content-Length: 0
*Mar  6 14:19:19: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.231:53600
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
To: "3660110" <sip:3660110@166.34.245.230>
Date: Sat, 06 Mar 1993 19:19:19 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Timestamp: 731612717
Content-Length: 0
CSeq: 101 BYE
```

The following example show debug output from the other side of the call:

```
Router2# debug ccsip messages
SIP Call messages tracing is enabled
Router2#
*Mar  8 17:45:12: Received:
INVITE sip:3660210@166.34.245.231;user=phone;phone-context=unknown SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Sat, 06 Mar 1993 19:19:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Cisco-Guid: 2881152943-2184249568-0-483551624
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731427554
Contact: <sip:3660110@166.34.245.230:5060;user=phone>
Expires: 180
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 5596 7982 IN IP4 166.34.245.230
s=SIP Call
t=0 0
```

```
c=IN IP4 166.34.245.230
m=audio 20762 RTP/AVP 0
*Mar  8 17:45:12: Sent:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Length: 0
*Mar  8 17:45:12: Sent:
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 1193 7927 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20224 RTP/AVP 0
*Mar  8 17:45:14: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Contact: <sip:3660210@166.34.245.231:5060;user=phone>
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 1193 7927 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20224 RTP/AVP 0
*Mar  8 17:45:14: Received:
ACK sip:3660210@166.34.245.231:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:55820
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
Date: Sat, 06 Mar 1993 19:19:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Max-Forwards: 6
Content-Type: application/sdp
Content-Length: 138
CSeq: 101 ACK
v=0
o=CiscoSystemsSIP-GW-UserAgent 5596 7982 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20762 RTP/AVP 0
*Mar  8 17:45:17: Sent:
BYE sip:3660110@166.34.245.230:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.231:53600
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
To: "3660110" <sip:3660110@166.34.245.230>
Date: Mon, 08 Mar 1993 22:45:14 GMT
```

```
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 6
Timestamp: 731612717
CSeq: 101 BYE
Content-Length: 0
*Mar  8 17:45:17: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.231:53600
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
To: "3660110" <sip:3660110@166.34.245.230>
Date: Sat, 06 Mar 1993 19:19:19 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Timestamp: 731612717
Content-Length: 0
CSeq: 101 BYE
```

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging.
debug ccsip calls	Shows all SIP SPI call tracing.
debug ccsip error	Shows SIP SPI errors.
debug ccsip events	Shows all SIP SPI events tracing.
debug ccsip states	Shows all SIP SPI state tracing.

debug ccsip preauth

To enable diagnostic reporting of authentication, authorization, and accounting (AAA) preauthentication for Session Initiation Protocol (SIP) calls, use the **debugccsippreauth** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccsip preauth

no debug ccsip preauth

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced.

Examples

The following example shows debug output for a single SIP call:

```
Router# debug ccsip preauth
SIP Call preauth tracing is enabled
Jan 23 18:43:17.898::Preauth Required
Jan 23 18:43:17.898: In sipSPISendPreauthReq for preauth_id = 86515, ccb = 67AF4E10
Jan 23 18:43:17.898: Entering rpms_proc_print_preauth_req
Jan 23 18:43:17.898: Request = 0
Jan 23 18:43:17.898: Preauth id = 86515
Jan 23 18:43:17.898: EndPt Type  = 1
Jan 23 18:43:17.898: EndPt = 192.168.80.70
Jan 23 18:43:17.898: Resource Service = 1
Jan 23 18:43:17.898: Call_origin = answer
Jan 23 18:43:17.898: Call_type = voip
Jan 23 18:43:17.898: Calling_num = 2270001
Jan 23 18:43:17.898: Called_num = 1170001
Jan 23 18:43:17.898: Protocol  = 1
Jan 23 18:43:17.898:sipSPISendPreauthReq:Created node with preauth_id = 86515, ccb 67AF4E10
 , node 6709C280
Jan 23 18:43:17.898:rpms_proc_create_node:Created node with preauth_id = 86515
Jan 23 18:43:17.898:rpms_proc_send_aaa_req:uid got is 466728
Jan 23 18:43:17.902:rpms_proc_preauth_response:Context is for preauth_id 86515, aaa_uid
466728
Jan 23 18:43:17.902:rpms_proc_preauth_response:Deleting Tree node for preauth id 86515 uid
 466728
Jan 23 18:43:17.902:sipSPIGetNodeForPreauth:Preauth_id=86515
Jan 23 18:43:17.902: ccsip_spi_process_preauth_event:67AF4E10 ccb & 6709C280 node
Jan 23 18:43:17.902: In act_preauth_response:67AF4E10 ccb
Jan 23 18:43:17.902: act_preauth_response:Deleting node 6709C280 from tree
```

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

Table 17: debug ccsip preauth Field Descriptions

Field	Description
Request	Request Type--0 for preauthentication, 1 for disconnect.
Preauth id	Identifier for the preauthentication request.
EndPt Type	Call Origin End Point Type--1 for IP address, 2 for Interzone ClearToken (IZCT) value.
EndPt	Call Origin End Point Value--An IP address or IZCT value.
Resource Service	Resource Service Type--1 for Reservation, 2 for Query.
Call_origin	Answer.
Call_type	Voice over IP (VoIP).
Calling_num	Calling Party Number (CLID).
Called_num	Called Party Number (DNIS).
Protocol	0 for H.323, 1 for SIP.
function reports	Various identifiers and status reports for executed functions.

debug ccsip states

To show all Session Initiation Protocol (SIP) Service Provider Interface (SPI) state tracing, use the **debugccsipstates** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccsip states

no debug ccsip states

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.2(2)XA	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB1	This command was implemented on the Cisco AS5850 universal gateway.
12.2(8)T	This command was implemented on Cisco 7200 series routers.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T. Support for the Cisco AS5300 universal access server, Cisco AS5350, Cisco AS5400, and Cisco AS5850 universal gateway is not included in this release.
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

This command traces the state machine changes of SIP SPI and displays the state transitions.

Examples

The following example shows all SIP SPI state tracing:

```
Router1# debug ccsip states
SIP Call states tracing is enabled
Router1#
*Jan 2 18:34:37.793:0x6220C634 :State change from (STATE_NONE, SUBSTATE_NONE) to (STATE_IDLE,
 SUBSTATE_NONE)
*Jan 2 18:34:37.797:0x6220C634 :State change from (STATE_IDLE, SUBSTATE_NONE) to (STATE_IDLE,
 SUBSTATE_CONNECTING)
```

```
*Jan 2 18:34:37.797:0x6220C634 :State change from (STATE_IDLE, SUBSTATE_CONNECTING) to
(STATE_IDLE, SUBSTATE_CONNECTING)
*Jan 2 18:34:37.801:0x6220C634 :State change from (STATE_IDLE, SUBSTATE_CONNECTING) to
(STATE_SENT_INVITE, SUBSTATE_NONE)
*Jan 2 18:34:37.809:0x6220C634 :State change from (STATE_SENT_INVITE, SUBSTATE_NONE) to
(STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROCEEDING)
*Jan 2 18:34:37.853:0x6220C634 :State change from (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_PROCEEDING) to (STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_ALERTING)
*Jan 2 18:34:38.261:0x6220C634 :State change from (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_ALERTING) to (STATE_ACTIVE, SUBSTATE_NONE)
*Jan 2 18:35:09.860:0x6220C634 :State change from (STATE_ACTIVE, SUBSTATE_NONE) to
(STATE_DISCONNECTING, SUBSTATE_NONE)
*Jan 2 18:35:09.868:0x6220C634 :State change from (STATE_DISCONNECTING, SUBSTATE_NONE) to
(STATE_DEAD, SUBSTATE_NONE)
*Jan 2 18:28:38.404: Queued event from SIP SPI :SIPSPI_EV_CLOSE_CONNECTION
```

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging.
debug ccsip calls	Shows all SIP SPI call tracing.
debug ccsip error	Shows SIP SPI errors.
debug ccsip events	Shows all SIP SPI events tracing.
debug ccsip info	Shows all SIP SPI message tracing.

debug ccsip transport

To enable tracing of the Session Initiation Protocol (SIP) transport handler and the TCP or User Datagram Protocol (UDP) process, use the **debugccsiptransport**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccsip transport

no debug ccsip transport

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2 SX	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 the **debugccsiptransport**command to debug issues related to connection and transport usage and to see the flow of the messages being sent or received.

Examples

The following is sample output from the **debugccsiptransport**command for a Cisco 3660:

```
Router# debug ccsip transport
.
.
.
1w1d: //18/8E16980D800A/SIP/Transport/sipSPISendInvite: Sending Invite to the transport
layer
1w1d: //18/8E16980D800A/SIP/Transport/sipSPIGetSwitchTransportFlag: Return the Global
configuration, Switch Transport is TRUE
1w1d: //18/8E16980D800A/SIP/Transport/sipSPITransportSendMessage: msg=0x64082D50,
addr=172.18.194.183, port=5060, sentBy_port=0, is_req=1, transport=1, switch=1,
callBack=0x614FAB58
1w1d: //18/8E16980D800A/SIP/Transport/sipSPITransportSendMessage: Proceedable for sending
msg immediately
1w1d: //18/8E16980D800A/SIP/Transport/sipTransportLogicSendMsg: switch transport is 1
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportGetInterfaceMtuSize: MTU size for remote
 address 172.18.194.183 is 500
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportVerifyMsgForMTUThreshold: Interface MTU
Size 500, Msg Size 1096
1w1d: //18/8E16980D800A/SIP/Transport/sipTransportLogicSendMsg: Switching msg=0x64082D50
transport UDP->TCP
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportSetAgeingTimer: Aging timer initiated for
 holder=0x64084058,addr=172.18.194.183
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipCreateConnHolder: Created new holder=0x64084058,
addr=172.18.194.183
```

```
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportPostRequestConnection: Posting TCP conn
create request for addr=172.18.194.183, port=5060, context=0x64128D5C
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportSetConnWaitTimer: Wait timer set for
connection=0x64129BF4,addr=172.18.194.183, port=5060
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipCreateConnInstance: Created new initiated
conn=0x64129BF4, connid=-1, addr=172.18.194.183, port=5060, transport=tcp
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipConnectionManagerProcessConnCreated:
gConnTab=0x64128D5C, addr=172.18.194.183, port=5060, connid=1, transport=tcp
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipInstanceHandleConnectionCreated: Moving
connection=0x64129BF4, connid=1state to pending
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportProcessNWConnectionCreated:
context=0x64128D5C
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipConnectionManagerProcessConnCreated:
gConnTab=0x64128D5C, addr=172.18.194.183, port=5060, connid=1, transport=tcp
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportPostSendMessage: Posting send for
msg=0x64082D50, addr=172.18.194.183, port=5060, connId=1 for TCP
.
.
.
```

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

Table 18: debug ccsip transport Field Descriptions

Field	Description
Sending Invite to the transport layer	Indicates that the SIP signaling state machine has invoked transport layer operations such as transport arbitration logic and the connection management interface.
switch transport is 1	Indicates that the gateway has been provisioned to enable the transport switching functionality based on the message size. 1 is true and 0 is false.
MTU size for remote address	Indicates that the bound outgoing Ethernet interface that sends the message to the given remote address is configured for an MTU size of the indicated value.
Interface MTU Size 500, Msg Size 1096	Indicates that the size of the message is larger than the size of the MTU; thus transport switching (from UDP to TCP) should be enabled.
Switching msg=... transport UDP->TCP	Indicates that transport switching from UDP to TCP is occurring for the handled message because of the large size of the message.
Aging timer initiated for holder	Indicates that the connection algorithm is started; that is, the counter begins to age out the TCP or UDP connection if inactivity occurs.
Posting TCP conn create request	Indicates a request for a TCP connection from a lower TCP process.

Field	Description
sipSPITransportSendMessage:msg=0x64082D50, addr=...transport=1, switch=1, callBack=0x614FAB58	Indicates all the transport related attributes that the SIP signaling state machine originally gives to the transport layer to send out the message. The attributes are: • transport: 1 for UDP; 2 for TCP. • switch (switching transport enabled or disabled for large messages): 1 for enabled; 0 for disabled.
Posting send for msg=0x64082D50, addr=...for TCP	Indicates that all transport and connection related operations are complete. The message is sent out on the network targeted to the given address, port, and transport.

Related Commands

Command	Description
debug ccsip all	Enables all SIP-related debugging.
debug ccsip info	Enables tracing of general SIP SPI information.
transport switch	Enables switching between UDP and TCP transport mechanisms globally for large SIP messages.
voice-class sip transport switch	Enables switching between UDP and TCP transport mechanisms for large SIP messages for a specific dial peer.

debug ccswvoice vo-debug

To display detailed debugging information related to ccswvoice function calls during call setup and teardown, use the **debugccswvoicevo-debug** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccswvoice vo-debug

no debug ccswvoice vo-debug

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(1)MA	This command was introduced on the Cisco MC3810 networking device.
12.0(7)XK	This command was implemented on the Cisco 3600 series router.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.

Usage Guidelines

Use this command when attempting to troubleshoot a Vo call that uses the "cisco-switched" session protocol. This command provides the same information as the **debugccswvoicevo-session** command, but includes additional debugging information relating to the calls.

Examples

The following shows sample output from the **debugccswvoicevo-debug** command:

```
Router# debug ccswvoice vo-debug
2w2d: ccswvoice: callID 529927 pvcid -1 cid -1 state NULL event O/G SETUP
2w2d: ccswvoice_out_callinit_setup: callID 529927 using pvcid 1 cid 15
2w2d: ccswvoice: callID 529927 pvcid 1 cid 15 state O/G INIT event I/C PROC
2w2d: ccswvoice: callID 529927 pvcid 1 cid 15 state O/G PROC event I/C ALERTccfrf11_caps_ind:
 codec(preferred) = 1

2w2d: ccswvoice: callID 529927 pvcid 1 cid 15 state O/G ALERT event I/C CONN
2w2d: ccswvoice_bridge_drop: dropping bridge calls src 529927 dst 529926 pvcid 1 cid 15
state ACTIVE
2w2d: ccswvoice: callID 529927 pvcid 1 cid 15 state ACTIVE event O/G REL
2w2d: ccswvoice: callID 529927 pvcid 1 cid 15 state RELEASE event I/C RELCOMP
2w2d: ccswvo_store_call_history_entry: cause=10 tcause=10    cause_text=normal call clearing.
```

Related Commands

Command	Description
debug ccswvoice vo-session	Displays the first 10 bytes (including header) of selected VoFR subframes for the interface .

debug ccswvoice vofr-debug

To display the ccswvoice function calls during call setup and teardown, use the **debugccswvoicevofr-debug** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccswvoice vofr-debug

no debug ccswvoice vofr-debug

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)XG	This command was introduced on the Cisco 2600 and Cisco 3600 series routers.
12.0(4)T	This command was integrated into Cisco IOS Release 12.0(4)T.
12.0(7)XK	This command was implemented on the Cisco MC3810 networking device.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.

Usage Guidelines

Use this command when troubleshooting a VoFR call that uses the "cisco-switched" session protocol. This command provides the same information as the **debugccswvoicevofr-session** command, but includes additional debugging information relating to the calls.

Examples

The following shows sample output from the **debugccswvoicevofr-debug** command:

```
Router# debug ccswvoice vofr-debug
CALL TEARDOWN:
3640_vofr(config-voiceport)#
*Mar  1 03:02:08.719:ccswvofr_bridge_drop:dropping bridge calls src 17 dst 16 dlci 100
      cid 9 state ACTIVE
*Mar  1 03:02:08.727:ccswvofr:callID 17 dlci 100 cid 9 state ACTIVE event O/G REL
*Mar  1 03:02:08.735:ccswvofr:callID 17 dlci 100 cid 9 state RELEASE event I/C RELCOMP
*Mar  1 03:02:08.735:ccswvofr_store_call_history_entry:cause=22 tcause=22
      cause_text=no circuit.
3640_vofr(config-voiceport)#
CALL SETUP (outgoing):
*Mar  1 03:03:22.651:ccswvofr:callID 23 dlci -1 cid -1 state NULL event O/G SETUP
*Mar  1 03:03:22.651:ccswvofr_out_callinit_setup:callID 23 using dlci 100 cid 10
*Mar  1 03:03:22.659:ccswvofr:callID 23 dlci 100 cid 10 state O/G INIT event I/C PROC
*Mar  1 03:03:22.667:ccswvofr:callID 23 dlci 100 cid 10 state O/G PROC event I/C CONN
ccfrf11_caps_ind:codec(preferred) = 0
```

Related Commands

Command	Description
debug cch323	Displays the ccfrf11 function calls during call setup and teardown.
debug ccsw voice vo-debug	Displays the ccswvoice function calls during call setup and teardown.
debug vtsp session	Displays the first 10 bytes (including header) of selected VoFR subframes for the interface.

debug ccswvoice vofr-session

To display the ccswvoice function calls during call setup and teardown, use the **debugccswvoicevofr-session** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccswvoice vofr-session

no debug ccswvoice vofr-session

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)XG	This command was introduced on the Cisco 2600 and Cisco 3600 series routers.
12.0(4)T	This command was integrated into Cisco IOS Release 12.0(4)T.
12.0(7)XK	This command was implemented on the Cisco MC3810 networking device.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.

Usage Guidelines

Use this command to show the state transitions of the cisco-switched-vofr state machine as a call is processed, and when attempting to troubleshoot a VoFR call that uses the "cisco-switched" session protocol.

Examples

The following shows sample output from the **debugccswvoicevofr-session** command:

```
Router# debug ccswvoice vofr-session
CALL TEARDOWN:
3640_vofr(config-voiceport)#
*Mar  1 02:58:13.203:ccswvofr:callID 14 dlci 100 cid 8 state ACTIVE event O/G REL
*Mar  1 02:58:13.215:ccswvofr:callID 14 dlci 100 cid 8 state RELEASE event I/C RELCOMP
3640_vofr(config-voiceport)#
CALL SETUP (outgoing):
*Mar  1 02:59:46.551:ccswvofr:callID 17 dlci -1 cid -1 state NULL event O/G SETUP
*Mar  1 02:59:46.559:ccswvofr:callID 17 dlci 100 cid 9 state O/G INIT event I/C PROC
*Mar  1 02:59:46.567:ccswvofr:callID 17 dlci 100 cid 9 state O/G PROC event I/C CONN
3640_vofr(config-voiceport)#
```

Related Commands

Command	Description
debug cch323	Displays the ccfrf11 function calls during call setup and teardown.

Command	Description
debug call rsvp-sync events	Displays events that occur during RSVP setup.
debug vtsp session	Displays the first 10 bytes (including header) of selected VoFR subframes for the interface.

debug ccswvoice vo-session

To display the first 10 bytes (including header) of selected VoFR subframes for the interface , use the **debugccswvoicevo-session** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ccswvoice vo-session

no debug ccswvoice vo-session

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(1)MA	This command was introduced on the Cisco MC3810 networking device.
12.0(7)XK	This command was implemented on the Cisco 3600 series router.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.

Usage Guidelines

Use this command to show the state transitions of the cisco-switched-vo state machine as a call is processed. This command should be used when attempting to troubleshoot a Vo call that uses the "cisco-switched" session protocol.

Examples

The following shows sample output from the **debugccswvoicevo-session** command:

```
Router# debug ccswvoice vo-session
2w2d: ccswvoice: callID 529919 pvcid -1 cid -1 state NULL event O/G SETUP
2w2d: ccswvoice: callID 529919 pvcid 1 cid 11 state O/G INIT event I/C PROC
2w2d: ccswvoice: callID 529919 pvcid 1 cid 11 state O/G PROC event I/C ALERT
2w2d: ccswvoice: callID 529919 pvcid 1 cid 11 state O/G ALERT event I/C CONN
2w2d: ccswvoice: callID 529919 pvcid 1 cid 11 state ACTIVE event O/G REL
2w2d: ccswvoice: callID 529919 pvcid 1 cid 11 state RELEASE event I/C RELCOMP
```

Related Commands

Command	Description
debug ccswvoice vo-debug	Displays detailed debugging information related to ccswvoice function calls during call setup and teardown.

debug cdapi

To display information about the Call Distributor Application Programming Interface (CDAPI), use the **debugcdapi**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdapi {**detail**| **events**}

no debug cdapi {**detail**| **events**}

Syntax Description

detail	Displays when applications register or become unregistered with CDAPI, when calls are added or deleted from the CDAPI routing table, and when CDAPI messages are created and freed.
events	Displays the events passing between CDAPI and an application or signalling stack.

Command Default

Debugging output is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(6)T	This command was introduced.
12.1(5)XM2	This command was implemented on the Cisco AS5350 and Cisco AS5400.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T. This command was enhanced to show V.110 call types.
12.3(4)T	This command was enhanced to show V.120 call types.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The **detail** keyword is useful for determining if messages are being lost (or not freed). It is also useful for determining the size of the raw messages passed between CDAPI and other applications to ensure that the correct number of bytes is being passed.

The **events** keyword is useful for determining if certain ISDN messages are not being received by an application and if calls are not being directed to an application.

The following bandwidths are supported:

- 56 kbps
- 64 kbps

Examples

The following Media Gateway Control Protocol (MGCP) packet received example shows V.110 call debugging output for the **debugcdapidetail** command. In this example, the modem is not yet in STEADY_STATE.

```
Router# debug cdapi detail
Sep 26 19:12:25.327:MGCP Packet received from 10.0.44.109:2427-
CRCX 6318 s7/ds1-0/24 MGCP 1.0
C:111
M:nas/data
L:b:64, nas/bt:v.110, nas/cdn:234567
R:nas/au, nas/ax,nas/of, nas/crq
X:101
Sep 26 19:12:25.327:CDAPI:cdapi_create_msg():CDAPI Pool Count:959, Raw Length = 0
Sep 26 19:12:25.327:CDAPI Se7/1:23:cdapi_add_entry_callRoutingTbl() -
Sep 26 19:12:25.327:   Added entry for call 0x7017 for application CSM
Sep 26 19:12:25.331:CDAPI:cdapi_create_msg():CDAPI Pool Count:958,
router# Raw Length = 0
Sep 26 19:12:25.331:CDAPI:cdapi_free_msg():Raw Length = 0, freeRaw = 1, Raw Msg = 0x0
Sep 26 19:12:25.331:CDAPI:cdapi_free_msg():CDAPI Pool Count:959
Sep 26 19:12:25.331:CDAPI:cdapi_free_msg():Raw Length = 0, freeRaw = 1, Raw Msg = 0x0
Sep 26 19:12:25.331:CDAPI:cdapi_free_msg():CDAPI Pool Count:960
Sep 26 19:12:25.331:send_mgcp_msg, MGCP Packet sent to 10.0.44.109:2427 --->
Sep 26 19:12:25.331:200 6318  Alert
I:64524608
Sep 26 19:12:25.339:CDAPI:cdapi_crea
router#te_msg():CDAPI Pool Count:959, Raw Length = 0
Sep 26 19:12:25.339:CDAPI:cdapi_free_msg():Raw Length = 0, freeRaw = 1, Raw Msg = 0x0
Sep 26 19:12:25.339:CDAPI:cdapi_free_msg():CDAPI Pool Count:960
router#
Sep 26 19:12:33.223:MGCP Packet received from 10.0.44.109:2427-
DLCX 6319 s7/ds1-0/24 MGCP 1.0
Sep 26 19:12:33.223:CDAPI:cdapi_create_msg():CDAPI Pool Count:959, Raw Length = 0
Sep 26 19:12:33.223:CDAPI:cdapi_create_msg():CDAPI Pool Count:958, Raw Length = 0
Sep 26 19:12:33.223:CDAPI:cdapi_free_msg():Raw Length = 0, freeRaw = 1, Raw Msg = 0x0
Sep 26 19:12:33.223:CDAPI:cdapi_free_msg():CDAPI Pool Count:959
Sep 26 19:12:33.227:CDAPI:cdapi_create_msg():CDAPI Pool Count:958, Raw
router# Length = 0
Sep 26 19:12:33.227:CDAPI Se7/1:23:cdapi_del_entry_callRoutingTbl() -
Sep 26 19:12:33.227:   Deleted entry for call 0x7017
Sep 26 19:12:33.227:CDAPI:cdapi_free_msg():Raw Length = 0, freeRaw = 1, Raw Msg = 0x0
Sep 26 19:12:33.227:CDAPI:cdapi_free_msg():CDAPI Pool Count:959
Sep 26 19:12:33.227:CDAPI:cdapi_free_msg():Raw Length = 0, freeRaw = 1, Raw Msg = 0x0
Sep 26 19:12:33.227:CDAPI:cdapi_free_msg():CDAPI Pool Count:960
Sep 26 19:12:33.227:send_mgcp_msg, MGCP Packet sent
router#to 10.0.44.109:2427 --->
Sep 26 19:12:33.227:200 6319 OK
```

The following partial example shows V.120 call debugging output for the **debugcdapidetail** command:

```
Router# debug cdapi detail
May 14 19:12:25.327:MGCP Packet received from 10.0.44.109:2427-
CRCX 6318 s7/ds1-0/24 MGCP 1.0
C:111
M:nas/data
L:b:64, nas/bt:v.120, nas/cdn:234567
R:nas/au, nas/ax,nas/of, nas/crq
X:101
May 14 19:12:25.327:CDAPI:cdapi_create_msg():CDAPI Pool Count:959, Raw Length = 0
May 14 19:12:25.327:CDAPI Se7/1:23:cdapi_add_entry_callRoutingTbl() -
May 14 19:12:25.327:   Added entry for call 0x7017 for application CSM
May 14 19:12:25.331:CDAPI:cdapi_create_msg():CDAPI Pool Count:958,
router# Raw Length = 0
May 14 19:12:25.331:CDAPI:cdapi_free_msg():Raw Length = 0, freeRaw = 1, Raw Msg = 0x0
May 14 19:12:25.331:CDAPI:cdapi_free_msg():CDAPI Pool Count:959
May 14 19:12:25.331:CDAPI:cdapi_free_msg():Raw Length = 0, freeRaw = 1, Raw Msg = 0x0
```

```
May 14 19:12:25.331:CDAPI:cdapi_free_msg():CDAPI Pool Count:960
May 14 19:12:25.331:send_mgcp_msg, MGCP Packet sent to 10.0.44.109:2427 --->
.
.
.
```

The following MGCP packet received example shows V.120 call debugging output for the **debugcdapievents** command:

```
Router# debug cdapi events
Sep 26 19:14:39.027:MGCP Packet received from 10.0.44.109:2427-
CRCX 6322 s7/ds1-0/24 MGCP 1.0
C:111
M:nas/data
L:b:64, nas/bt:v.120, nas/cdn:234567
R:nas/au, nas/ax,nas/of, nas/crq
X:101
Sep 26 19:14:39.027:Se7/0:23 CDAPI:TX -> CDAPI_MSG_CONNECT_IND to CSM call = 0x7017
Sep 26 19:14:39.027:   From Appl/Stack = XCSP
Sep 26 19:14:39.027:   Call Type    = V.120
Sep 26 19:14:39.027:   B Channel    = 23
Sep 26 19:14:39.027:   dslId        = 0
Sep 26 19:14:39.027:   Idb          = 0
Sep
router#26 19:14:39.027:   BChanIdb  = 64519A14
Sep 26 19:14:39.027:   Handle       = 63CB8DF4
Sep 26 19:14:39.027:   RPA          = 6388506C
Sep 26 19:14:39.027:   Cause        = 0
Sep 26 19:14:39.027:   ApplCause    = 0
Sep 26 19:14:39.027:   ApplSpecData = 0
Sep 26 19:14:39.027:   Calling Party Number =
Sep 26 19:14:39.027:   Called Party Number = 234567
Sep 26 19:14:39.027:   Overlap = 0
Sep 26 19:14:39.027:Se7/0:23 CDAPI:TX -> CDAPI_MSG_CONNECT_RESP to XCSP call = 0x7017
Sep 26 19:14:39.027:   From Appl
router#/Stack = CSM
Sep 26 19:14:39.027:   Call Type    = MODEM
Sep 26 19:14:39.027:   B Channel    = 23
Sep 26 19:14:39.027:   dslId        = 0
Sep 26 19:14:39.027:   Idb          = 0
Sep 26 19:14:39.027:   BChanIdb     = 64519A14
Sep 26 19:14:39.027:   Handle       = 63CB8DF4
Sep 26 19:14:39.027:   RPA          = 0
Sep 26 19:14:39.027:   Cause        = 0
Sep 26 19:14:39.027:   ApplCause    = 0
Sep 26 19:14:39.027:   ApplSpecData = 0
Sep 26 19:14:39.027:   Overlap = 0
Sep 26 19:14:39.031:send_mgcp_msg, MGCP Pa
router#cket sent to 10.0.44.109:2427 --->
Sep 26 19:14:39.031:200 6322  Alert
I:64524608
Sep 26 19:14:39.039:Se7/0:23 CDAPI:TX -> CDAPI_MSG_CONN_ACT_REQ to XCSP call = 0x7017
Sep 26 19:14:39.039:   From Appl/Stack = CSM
Sep 26 19:14:39.039:   Call Type    = MODEM
Sep 26 19:14:39.039:   B Channel    = 23
Sep 26 19:14:39.039:   dslId        = 0
Sep 26 19:14:39.039:   Idb          = 0
Sep 26 19:14:39.039:   BChanIdb     = 64519A14
Sep 26 19:14:39.039:   Handle       = 63CB8DF4
Sep 26 19:14:39.039:   R
router#PA          = 0
Sep 26 19:14:39.039:   Cause        = 0
Sep 26 19:14:39.039:   ApplCause    = 0
Sep 26 19:14:39.039:   ApplSpecData = 0
Sep 26 19:14:39.039:   Overlap = 0
router#
Sep 26 19:14:48.959:MGCP Packet received from 10.0.44.109:2427-
DLCX 6323 s7/ds1-0/24 MGCP 1.0
Sep 26 19:14:48.963:Se7/0:23 CDAPI:TX -> CDAPI_MSG_DISCONNECT_IND to CSM call = 0x7017
Sep 26 19:14:48.963:   From Appl/Stack = XCSP
Sep 26 19:14:48.963:   Call Type    = V.120
Sep 26 19:14:48.963:   B Channel    = 23
```

```
Sep 26 19:14:48.963:   dslId        = 0
Sep 26 19:14:48.963:   Idb          = 0
Sep 26 19:14:48.963:   BChanIdb     = 64519A14
Sep 26 19:14:48.963:   Handle       = 63CB8DF4
Sep 26 19:14
router#:48.963:        RPA          = 6388506C
Sep 26 19:14:48.963:   Cause        = 0
Sep 26 19:14:48.963:   ApplCause    = 0
Sep 26 19:14:48.963:   ApplSpecData = 0
Sep 26 19:14:48.963:   Overlap = 0
Sep 26 19:14:48.963:Se7/0:23 CDAPI:TX -> CDAPI_MSG_SUBTYPE_RELEASE_REQ to XCSP call = 0x7017
Sep 26 19:14:48.963:   From Appl/Stack = CSM
Sep 26 19:14:48.963:   Call Type    = MODEM
Sep 26 19:14:48.963:   B Channel    = 23
Sep 26 19:14:48.963:   dslId        = 0
Sep 26 19:14:48.963:   Idb          = 0
Sep 26 19:14:48
router#.963:           BChanIdb     = 64519A14
Sep 26 19:14:48.963:   Handle       = 63CB8DF4
Sep 26 19:14:48.963:   RPA          = 0
Sep 26 19:14:48.963:   Cause        = 0
Sep 26 19:14:48.963:   ApplCause    = 1
Sep 26 19:14:48.963:   ApplSpecData = 0
Sep 26 19:14:48.963:   Overlap = 0
Sep 26 19:14:48.963:Se7/0:23 CDAPI:TX -> CDAPI_MSG_SUBTYPE_REL_COMP_IND to CSM call = 0x7017
Sep 26 19:14:48.963:   From Appl/Stack = XCSP
Sep 26 19:14:48.963:   Call Type    = V.120
Sep 26 19:14:48.963:   B Channel    = 23
Sep 26 19:
router#14:48.963:      dslId        = 0
Sep 26 19:14:48.963:   Idb          = 0
Sep 26 19:14:48.963:   BChanIdb     = 64519A14
Sep 26 19:14:48.963:   Handle       = 63CB8DF4
Sep 26 19:14:48.963:   RPA          = 6388506C
Sep 26 19:14:48.963:   Cause        = 0
Sep 26 19:14:48.963:   ApplCause    = 0
Sep 26 19:14:48.963:   ApplSpecData = 0
Sep 26 19:14:48.963:   Overlap = 0
Sep 26 19:14:48.963:send_mgcp_msg, MGCP Packet sent to 10.0.44.109:2427 --->
Sep 26 19:14:48.963:200 6323 OK
```

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

Table 19: debug cdapi Field Descriptions

Field	Description
L:b:64, nas/bt	The bearer type parameter includes v.110 and v.120 for V.110 and V.120 calls.
Call Type	Call types are V.110, V.120, and modem.

Related Commands

Command	Description
debug mgcp packet	Displays the MGCP signaling message received and sent to the called agent.
debug voip rawmsg	Displays the raw message owner, length, and pointer.

debug cdma pdsn a10 ahdlc

To display debug messages for Asynchronous High-Level Data Link Control (AHDLC), use the **debugcdmapdsna10ahdlc**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn a10 ahdlc [**errors**| **events**]

no debug cdma pdsn a10 ahdlc [**errors**| **events**]

Syntax Description

errors	(Optional) Displays details of AHDLC packets in error.
events	(Optional) Displays AHDLC events.

Command Default

If the command is entered without any optional keywords, all of the types of debug information are enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XC	This command was introduced.
12.2(8)BY	Keywords were made optional.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

The following is sample output from the **debugcdmapdsna10ahdlc**command:

```
Router# debug cdma pdsn a10 ahdlc errors

ahdlc error packet display debugging is on
Router# debug cdma pdsn a10 ahdlc events
ahdlc events display debugging is on
Router#
*Jan  1 00:18:30:%LINK-3-UPDOWN:Interface Virtual-Access1, changed state to up
*Jan  1 00:18:30:*****OPEN AHDLC*****
*Jan  1 00:18:30: ahdlc_mgr_channel_create
*Jan  1 00:18:30: ahdlc_mgr_allocate_available_channel:
*Jan  1 00:18:30:ahdlc:tell h/w open channel 9 from engine 0
```

debug cdma pdsn a10 gre

To display debug messages for A10 Generic Routing Encapsulation (GRE) interface errors, events, and packets, use the **debugcdmapdsna10gre**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn a10 gre [**errors**| **events**| **packets**] [**tunnel-key** *key*]

no debug cdma pdsn a10 gre [**errors**| **events**| **packets**]

Syntax Description

errors	(Optional) Displays A10 GRE errors.
events	(Optional) Displays A10 GRE events.
packets	(Optional) Displays transmitted or received A10 GRE packets.
tunnel-key *key*	(Optional) Specifies the GRE key.

Command Default

If the command is entered without any optional keywords, all of the types of debug information are enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.2(8)BY	The **tunnel-key** keyword was added and the existing keywords were made optional.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

The following is sample output from the **debugcdmapdsna10greeventstunnel-key**command:

```
Router# debug cdma pdsn a10 gre events tunnel-key 1

Router# show debug
CDMA:
  CDMA PDSN A10 GRE events debugging is on for tunnel key 1
PDSN#
*Mar  1 04:00:57.847:CDMA-GRE:CDMA-Ix1 (GRE/CDMA) created with src 5.0.0.2 dst 0.0.0.0
*Mar  1 04:00:57.847:CDMA-GRE:(in) found session 5.0.0.2-4.0.0.1-1
*Mar  1 04:00:59.863:CDMA-GRE:(in) found session 5.0.0.2-4.0.0.1-1
*Mar  1 04:00:59.863:CDMA-GRE:(in) found session 5.0.0.2-4.0.0.1-1
*Mar  1 04:01:01.879:CDMA-GRE:(in) found session 5.0.0.2-4.0.0.1-1
```

```
*Mar  1 04:01:01.879:CDMA-GRE:(in) found session 5.0.0.2-4.0.0.1-1
*Mar  1 04:01:03.899:CDMA-GRE:(in) found session 5.0.0.2-4.0.0.1-1
*Mar  1 04:01:03.899:CDMA-GRE:(in) found session 5.0.0.2-4.0.0.1-1
```

debug cdma pdsn a10 ppp

To display debug messages for A10 Point-to-Point protocol (PPP) interface errors, events, and packets, use the **debugcdmapdsna10ppp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn a10 ppp [**errors**| **events**| **packets**]

no debug cdma pdsn a10 ppp [**errors**| **events**| **packets**]

Syntax Description

errors	(Optional) Displays A10 PPP errors.
events	(Optional) Displays A10 PPP events.
packets	(Optional) Displays transmitted or received A10 PPP packets.

Command Default

If the command is entered without any optional keywords, all of the types of debug information are enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.2(8)BY	Keywords were made optional.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

The following is sample output from the **debugcdmapdsna10ppp**command:

```
Router# debug cdma pdsn a10 ppp errors
CDMA PDSN A10 errors debugging is on
Router# debug cdma pdsn a10 ppp events

CDMA PDSN A10 events debugging is on
Router# debug cdma pdsn a10 ppp packets

CDMA PDSN A10 packet debugging is on
Router# show debug

*Jan  1 00:13:09:CDMA-PPP:create_va tunnel=CDMA-Ix1 virtual-template
template=Virtual-Template2 ip_enabled=1
*Jan  1 00:13:09:CDMA-PPP:create_va va=Virtual-Access1
*Jan  1 00:13:09:CDMA-PPP:clone va=Virtual-Access1 subif_state=1 hwidb->state=0
*Jan  1 00:13:09:          linestate=1 ppp_lineup=0
```

```
*Jan  1 00:13:09:%LINK-3-UPDOWN:Interface Virtual-Access1, changed state to up
*Jan  1 00:13:09:CDMA-PPP:clone va=Virtual-Access1 subif_state=1 hwidb->state=4
*Jan  1 00:13:09:          linestate=0 ppp_lineup=0
*Jan  1 00:13:09:*****OPEN AHDLC*****
```

debug cdma pdsn a11

To display debug messages for A11 interface errors, events, and packets, use the **debugcdmapdsna11**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn a11 [**errors**| **events**| **packets**] [*mnid*]

no debug cdma pdsn a11 [**errors**| **events**| **packets**]

Syntax Description

errors	(Optional) Displays A11 protocol errors.
events	(Optional) Displays A11 events.
packets	(Optional) Displays transmitted or received packets.
mnid	(Optional) Specifies the ID of the mobile station.

Command Default

If the command is entered without any optional keywords, all of the types of debug information are enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.2(8)BY	The *mnid* argument was added and the existing keywords were made optional.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

The following is sample output from the **debugcdmapdsna11**commands:

```
Router# debug cdma pdsn a11 errors
CDMA PDSN A11 errors debugging is on
Router# show debug
1d21h:CDMA-RP:(in) rp_msgs, code=1, status=0
1d21h:CDMA-RP:(enqueue req) type=1 homeagent=5.0.0.2 coaddr=4.0.0.1
1d21h:                      id=0xBEF750F0-0xBA53E0F lifetime=65535
1d21h:CDMA-RP:len=8, 00-00-00-00-00-00-00-F1 convert to 00000000000001
(14 digits), type=IMSI
1d21h:CDMA-RP:(req) process_rp_req, homeagent=5.0.0.2 coaddr=4.0.0.1
1d21h:                lifetime=65535 id=BEF750F0-BA53E0F
imsi=00000000000001
1d21h:CDMA-RP:(req) rp_req_create, 5.0.0.2-4.0.0.1-1 imsi=00000000000001
1d21h:CDMA-RP:(out) rp_reply session=5.0.0.2-4.0.0.1-1, lifetime=65535
1d21h:CDMA-RP:(out) setup_rp_out_msg, ha=5.0.0.2 coa=4.0.0.1 key=1
```

```
1d21h:%LINK-3-UPDOWN:Interface Virtual-Access2000, changed state to up
1d21h:CDMA-RP:ipmobile_visitor add/delete=1, mn=8.0.2.132, ha=7.0.0.2
1d21h:%LINEPROTO-5-UPDOWN:Line protocol on Interface Virtual-Access2000,
changed state to up
Router# debug cdma pdsn a11 packets events
Router# show debug
CDMA:
  CDMA PDSN A11 packet debugging is on for mnid 000000000000001
  CDMA PDSN A11 events debugging is on for mnid 000000000000001
Router#
*Mar  1 03:15:32.507:CDMA-RP:len=8, 01-00-00-00-00-00-00-10 convert to 000000000000001 (15
 digits), type=IMSI
*Mar  1 03:15:32.511:CDMA-RP:extension type=38, len=0
*Mar  1 03:15:32.511:CDMA-RP:extension type=38, len=0
*Mar  1 03:15:32.511:CDMA-RP:extension type=38, len=0
*Mar  1 03:15:32.511:CDMA-RP:extension type=32, len=20
*Mar  1 03:15:32.511:         00 00 01 00 EE 1F FC 43 0A 7D F9 36 29 C2 BA 28
*Mar  1 03:15:32.511:         5A 64 D5 9C
*Mar  1 03:15:32.511:CDMA-RP:(req) process_rp_req, homeagent=5.0.0.2 coaddr=4.0.0.1
*Mar  1 03:15:32.511:               lifetime=1800 id=AF3BFE55-69A109D IMSI=000000000000001
*Mar  1 03:15:32.511:CDMA-RP:(req) rp_req_create, ha=5.0.0.2, coa=4.0.0.1, key=1
IMSI=000000000000001
*Mar  1 03:15:32.511:CDMA-RP:(out) rp_reply session=5.0.0.2-4.0.0.1-1, lifetime=1800
*Mar  1 03:15:32.511:CDMA-RP:(out) Setup RP out message, ha=5.0.0.2 coa=4.0.0.1 key=1
*Mar  1 03:15:38.555:CDMA-RP:simple ip visitor added, mn=9.2.0.1, ha=0.0.0.0
Router#
*Mar  1 03:15:54.755:CDMA-RP:len=8, 01-00-00-00-00-00-00-10 convert to 000000000000001 (15
 digits), type=IMSI
*Mar  1 03:15:54.755:CDMA-RP:extension type=38, len=0
*Mar  1 03:15:54.755:CDMA-RP:extension type=32, len=20
*Mar  1 03:15:54.755:         00 00 01 00 EA 9C C6 4C BA B9 F9 B6 DD C4 19 76
*Mar  1 03:15:54.755:         51 5A 56 45
*Mar  1 03:15:54.755:CDMA-RP:(req) process_rp_req, homeagent=5.0.0.2 coaddr=4.0.0.1
*Mar  1 03:15:54.755:               lifetime=0 id=AF3BFE6B-4616E475 IMSI=000000000000001
*Mar  1 03:15:54.755:CDMA-RP:(req) rp_req_lifetime_zero 5.0.0.2-4.0.0.1-1
*Mar  1 03:15:54.755:               IMSI=000000000000001
*Mar  1 03:15:54.755:CDMA-RP:(out) rp_reply session=5.0.0.2-4.0.0.1-1, lifetime=0
*Mar  1 03:15:54.755:CDMA-RP:(out) Setup RP out message, ha=5.0.0.2 coa=4.0.0.1 key=1
Router# debug cdma pdsn a11 event mnid 000000000000001
Router# show debug
CDMA:
  CDMA PDSN A11 events debugging is on for mnid 000000000000001
Router#
*Mar  1 03:09:34.339:CDMA-RP:len=8, 01-00-00-00-00-00-00-10 convert to 000000000000001 (15
 digits), type=IMSI
*Mar  1 03:09:34.339:CDMA-RP:(req) process_rp_req, homeagent=5.0.0.2 coaddr=4.0.0.1
*Mar  1 03:09:34.339:               lifetime=1800 id=AF3BFCEE-DC9FC751 IMSI=000000000000001
*Mar  1 03:09:34.339:CDMA-RP:(req) rp_req_create, ha=5.0.0.2, coa=4.0.0.1, key=1
IMSI=000000000000001
*Mar  1 03:09:34.339:CDMA-RP:(out) rp_reply session=5.0.0.2-4.0.0.1-1, lifetime=1800
*Mar  1 03:09:34.339:CDMA-RP:(out) Setup RP out message, ha=5.0.0.2 coa=4.0.0.1 key=1
*Mar  1 03:09:40.379:CDMA-RP:simple ip visitor added, mn=9.2.0.1, ha=0.0.0.0
Router#
close the session
Router#
*Mar  1 03:10:00.575:CDMA-RP:len=8, 01-00-00-00-00-00-00-10 convert to 000000000000001 (15
 digits), type=IMSI
*Mar  1 03:10:00.575:CDMA-RP:(req) process_rp_req, homeagent=5.0.0.2 coaddr=4.0.0.1
*Mar  1 03:10:00.575:               lifetime=0 id=AF3BFD09-18040319 IMSI=000000000000001
*Mar  1 03:10:00.575:CDMA-RP:(req) rp_req_lifetime_zero 5.0.0.2-4.0.0.1-1
*Mar  1 03:10:00.575:               IMSI=000000000000001
*Mar  1 03:10:00.575:CDMA-RP:(out) rp_reply session=5.0.0.2-4.0.0.1-1, lifetime=0
*Mar  1 03:10:00.575:CDMA-RP:(out) Setup RP out message, ha=5.0.0.2 coa=4.0.0.1 key=1
Router# debug cdma pdsn a11 packet mnid 000000000000001

Router# show debug

CDMA:
  CDMA PDSN A11 packet debugging is on for mnid 000000000000001
Router#
*Mar  1 03:13:37.803:CDMA-RP:extension type=38, len=0
*Mar  1 03:13:37.803:CDMA-RP:extension type=38, len=0
*Mar  1 03:13:37.803:CDMA-RP:extension type=38, len=0
```

```
*Mar  1 03:13:37.803:CDMA-RP:extension type=32, len=20
*Mar  1 03:13:37.803:         00 00 01 00 A8 5B 30 0D 4E 2B 83 FE 18 C6 9D C2
*Mar  1 03:13:37.803:         15 BF 5B 57
*Mar  1 03:13:51.575:CDMA-RP:extension type=38, len=0
*Mar  1 03:13:51.575:CDMA-RP:extension type=32, len=20
*Mar  1 03:13:51.575:         00 00 01 00 58 77 E5 59 67 B5 62 15 17 52 83 6D
*Mar  1 03:13:51.579:         DC 0A B0 5B
```

debug cdma pdsn accounting

To display debug messages for accounting events, use the **debugcdmapdsnaccounting** command in privileged EXEC mode. To disable debugging output, use the no form of this command.

debug cdma pdsn accounting

no cdma pdsn accounting

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

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

```
Router# debug cdma pdsn accounting

CDMA PDSN accounting debugging is on
Router#
*Jan  1 00:15:32:CDMA/ACCT:null vaccess in session_start
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[9]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[44] len:[3] 01   Processing Y1
*Jan  1 00:15:32:CDMA/ACCT:    Setup airlink record received
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[12]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[41] len:[6] 00 00 00 02 CDMA/ACCT:
Processing Y2
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[9]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[42] len:[3] 12 CDMA/ACCT: Processing Y3
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1F] len:[17] 30 30 30 30 30 30 30 30
 30 30 30 30 30 32      Processing A1
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[12]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[9] len:[6] 04 04 04 05   Processing D3
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[14]
*Jan  1 00:15:32:CDMA/ACCT:   VSA Vid:5535 type:[10] len:[8] 00 00 04 04 04 05   Processing
 D4
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[9]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[44] len:[3] 02   Processing Y1
*Jan  1 00:15:32:CDMA/ACCT:    Start airlink record received
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[12]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[41] len:[6] 00 00 00 02 CDMA/ACCT:
Processing Y2
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[9]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[42] len:[3] 13 CDMA/ACCT: Processing Y3
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[10]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[11] len:[4] 00 02   Processing E1
```

```
*Jan  1 00:15:32:CDMA/ACCT: Current Attribute type:0x[1A] len:[10]
*Jan  1 00:15:32:CDMA/ACCT:    VSA Vid:5535 type:[12] len:[4] 00 F1   Processing F1
```

debug cdma pdsn accounting flow

To display debug messages for accounting flow, use the **debugcdmapdsnaccountingflow**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn accounting flow

no debug cdma pdsn accounting flow

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XC	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

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

```
Router# debug cdma pdsn accounting flow

CDMA PDSN flow based accounting debugging is on
pdsn-6500#
01:59:40:CDMA-SM:cdma_pdsn_flow_acct_upstream sess id 1 flow type 0 bytes 100 addr 20.20.20.1
01:59:40:CDMA-SM:cdma_pdsn_flow_acct_downstream sess id 1 flow type 0 bytes 100 addr
20.20.20.1
```

debug cdma pdsn accounting time-of-day

To display the timer value, use the **debugcdmapdsnaccounting** time-of-day command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn accounting time-of-day

no debug cdma pdsn accounting time-of-day

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

The following is sample output from the **debugcdmapdsnaccountingtime-of-day**command:

```
Router# debug cdma pdsn accounting time-of-day

CDMA PDSN accounting time-of-day debugging is on
Feb 15 19:13:23.634:CDMA-TOD:Current timer expiring in 22 seconds
Feb 15 19:13:24.194:%SYS-5-CONFIG_I:Configured from console by console
Router#
Feb 15 19:13:45.635:CDMA-TOD:Timer expired...Rearming timer
Feb 15 19:13:45.635:CDMA-TOD:Gathering session info
Feb 15 19:13:45.635:CDMA-TOD:Found 0 sessions
```

debug cdma pdsn cluster

To display the error messages, event messages, and packets received, use the **debugcdmapdsn**cluster command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn cluster message [**error**| **events**| **packets**] **redundancy** [**error**| **events**| **packets**]

no debug cdma pdsn cluster message [**error**| **events**| **packets**] **redundancy** [**error**| **events**| **packets**]

Syntax Description

message	Displays cluster messages for errors, events and packets received.
redundancy	Displays redundancy information for errors, events, and sent or received packets.
error	Displays either cluster or redundancy error messages.
events	Displays either all cluster or all redundancy events.
packets	Displays all transmitted or received cluster or redundancy packets.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines

This debug is *only* allowed on PDSN c6-mz images, and helps to monitor prepaid information.

Examples

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

```
Router# debug cdma pdsn cluster ?
  message     Debug PDSN cluster controller messages
  redundancy  Debug PDSN cluster controller redundancy
```

debug cdma pdsn ipv6

To display IPV6 error or event messages, use the debug cdma pdsn IPV6 command in privileged EXEC mode. To disable debug messages, use the no form of this command.

debug cdma pdsn ipv6

no debug cdma pdsn ipv6

Syntax Description

There are no arguments or keywords for this command.

Command Default

No default behavior or values.

Command History

Release	Modification
12.3(14)YX	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

Usage Guidelines

The following example illustrates the **debugcdmapdsnipv6**command:

```
Router# debug cdma pdsn ipv6
```

debug cdma pdsn prepaid

To display debug messages about prepaid flow, use the **debugcdmapdsnprepaid** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn prepaid

no debug cdma pdsn prepaid

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(8)BY	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines

This debug is *only* allowed on PDSN c6-mz images, and helps to monitor prepaid information.

Examples

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

```
Router# debug cdma pdsn prepaid
*Mar  1 00:09:38.391: CDMA-PREPAID:  Initialized the authorization request
*Mar  1 00:09:38.391: CDMA-PREPAID:   Added username into A-V list
*Mar  1 00:09:38.391: CDMA-PREPAID:   Added CLID into A-V list
*Mar  1 00:09:38.391: CDMA-PREPAID:   Added session id for prepaid
*Mar  1 00:09:38.391: CDMA-PREPAID:   Added correlation id into A-V list
*Mar  1 00:09:38.391: CDMA-PREPAID:   Added auth reason for prepaid into A-V list
*Mar  1 00:09:38.391: CDMA-PREPAID:   Added USER_ID for prepaid
*Mar  1 00:09:38.391: CDMA-PREPAID:   Added service id for prepaid
*Mar  1 00:09:38.391: CDMA-PREPAID:   Built prepaid VSAs
*Mar  1 00:09:38.391: CDMA-PREPAID:   Sent the request to AAA
*Mar  1 00:09:38.391: CDMA-PREPAID:   Auth_reason: CRB_RSP_PEND_INITIAL_QUOTA
*Mar  1 00:09:38.395: CDMA-PREPAID:   Received prepaid response: status 2
*Mar  1 00:09:38.395: CDMA-PREPAID:   AAA authorised parms being processed
*Mar  1 00:09:38.395: CDMA-PREPAID:   Attr in Grp Prof: crb-entity-type
*Mar  1 00:09:38.395: (0x4B000000) CDMA/PREPAID: AAA_AT_CRB_ENTITY_TYPE
*Mar  1 00:09:38.395: (0x4B000000) CDMA/PREPAID: entity type returns 1
*Mar  1 00:09:38.395: CDMA-PREPAID:   Attr in Grp Prof: crb-duration
*Mar  1 00:09:38.395: (0x4B000000) CDMA/PREPAID: AAA_AT_CRB_DURATION
*Mar  1 00:09:38.395: (0x4B000000) CDMA/PREPAID: duration returns 120
*Mar  1 00:09:38.395: CDMA-PREPAID:   Retrieved attributes successfully
*Mar  1 00:09:38.395: CDMA-PREPAID:   Reset duration to 120, mn 9.3.0.1
*Mar  1 00:09:38.395: CDMA-PREPAID: : Started duration timer for 120 sec
```

debug cdma pdsn qos

To display debug messages about quality of service features, use the **debugcdmapdsnqos** command in privileged EXEC mode. To disable debug messages, use the **no** form of this command.

debug cdma pdsn qos

no debug cdma pdsn qos

Syntax Description

There are no arguments or keywords for this command.

Command Default

There are no default values for this command.

Command History

Release	Modification
12.3(8)XW	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

Examples

There are currently no sample outputs for this command.

debug cdma pdsn resource-manager

To display debug messages that help you monitor the resource-manager information, use the **debugcdmapdsnresource-manager**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn resource-manager [**error**| **events**]

no debug cdma pdsn resource-manager [**error**| **events**]

Syntax Description

errors	Displays Packet Data Service node (PDSN) resource manager errors.
events	Displays PDSN resource manager events.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(8)BY	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

The following is sample output from the **debugcdmapdsresource-manager**command:

```
Router# debug cdma pdsn resource-manager

   errors  CDMA PDSN resource manager errors
   events  CDMA PDSN resource manager events
```

debug cdma pdsn selection

To display debug messages for the intelligent Packet Data Serving Node (PDSN) selection feature, use the **debugcdmapdsnselection**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn selection {**errors| events| packets**}

no debug cdma pdsn selection {**errors| events| packets**}

Syntax Description

errors	Displays PDSN selection errors.
events	Displays PDSN selection events.
packets	Displays transmitted or received packets.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

The following is sample output from the **debugcdmapdsnselection**command with the keyword **events** specified:

```
Router# debug cdma pdsn selection events

CDMA PDSN selection events debugging is on
Router#
00:27:46: CDMA-PSL: Message(IN) pdsn 51.4.2.40 interface 70.4.2.40
00:27:46:                Keepalive 10
00:27:46:                Count 0
00:27:46:                Capacity 16000
00:27:46:                Weight 0
00:27:46:                Hostname 11 7206-PDSN-2
00:27:46: CDMA-PSL: Reset keepalive, pdsn 51.4.2.40 current 10 new 10
00:27:46: CDMA-PSL: Message processed, pdsn 51.4.2.40 tsize 0 pendings 0
00:27:47: CDMA-PSL: Send KEEPALIVE, len 32
00:27:47: CDMA-PSL: Message(OUT) dest 224.0.0.11
00:27:47:                Keepalive 10
00:27:47:                Count 1
00:27:47:                Capacity 16000
00:27:47:                Weight 0
```

```
00:27:47:                Hostname 11 7206-PDSN-1
00:27:47: CDMA-PSL: RRQ sent, s=70.4.1.40 (FastEthernet0/1), d=224.0.0.11
```

debug cdma pdsn service-selection

To display debug messages for service selection, use the **debugcdmapdsnservice-selection**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn service-selection

no debug cdma pdsn service-selection

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

The following is sample output from the **debugcdmapdsnservice-selection** command:

```
Router# debug cdma pdsn service-selection
CDMA PDSN service provisioning debugging is on
Router#
1d02h:%LINK-3-UPDOWN:Interface Virtual-Access3, changed state to up
1d02h:Vi3 CDMA-SP:user_class=1, ms_ipaddr_req=1, apply_acl=0
1d02h:Vi3 CDMA-SP:Adding simple ip flow, user=bsip, mn=6.0.0.2,
1d02h:%LINEPROTO-5-UPDOWN:Line protocol on Interface Virtual-Access3,
changed state to up
```

debug cdma pdsn session

To display debug messages for Session Manager errors, events, and packets, use the **debugcdmapdsnsession**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdma pdsn session [**errors**| **events**]

no debug cdma pdsn session [**errors**| **events**]

Syntax Description

errors	(Optional) Displays session protocol errors.
events	(Optional) Displays session events.

Command Default

If the command is entered without any optional keywords, all of the types of debug information are enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XS	This command was introduced.
12.2(8)BY	Keywords were made optional.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Examples

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

```
Router# debug cdma pdsn session events

CDMA PDSN session events debugging is on
Router# debug cdma pdsn session errors

CDMA PDSN session errors debugging is on
Router# show debug

CDMA:
  CDMA PDSN session events debugging is on
  CDMA PDSN session errors debugging is on
Router#
*Jan  1 00:22:27:CDMA-SM:create_session 5.5.5.5-4.4.4.5-2
*Jan  1 00:22:27:CDMA-SM:create_tunnel 5.5.5.5-4.4.4.5
*Jan  1 00:22:27:%LINK-3-UPDOWN:Interface Virtual-Access1, changed state to up
*Jan  1 00:22:29:CDMA-SM:create_flow mn=0.0.0.0, ha=8.8.8.8 nai=l2tp2@cisco.com
*Jan  1 00:22:30:%LINEPROTO-5-UPDOWN:Line protocol on Interface Virtual-Access1, changed
state to up
```

debug cdp

To enable debugging of the Cisco Discovery Protocol (CDP), use the **debugcdp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdp {**packets**| **adjacency**| **events**}

no debug cdp {**packets**| **adjacency**| **events**}

Syntax Description

packets	Enables packet-related debugging output.
adjacency	Enables adjacency-related debugging output.
events	Enables output related to error messages, such as detecting a bad checksum.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(2)T	This command was introduced in a release earlier than Cisco IOS Release 12.4(2)T.
12.2(55)SE	This command was modified. The debug output was enhanced to display location Type-Length-Values (TLVs), location-server TLVs, and application TLV-related debugs.

Usage Guidelines

Use **debugcdp** commands to display information about CDP packet activity, activity between CDP neighbors, and various CDP events.

Examples

The following is sample output from the **debugcdppackets**,**debugcdpadjacency**,and**debugcdpevents** commands:

```
Router# debug cdp packets
CDP packet info debugging is on
Router# debug cdp adjacency
CDP neighbor info debugging is on
Router# debug cdp events
CDP events debugging is on
CDP-PA: Packet sent out on Ethernet0
CDP-PA: Packet received from gray.cisco.com on interface Ethernet0
CDP-AD: Deleted table entry for violet.cisco.com, interface Ethernet0
CDP-AD: Interface Ethernet2 coming up
CDP-EV: Encapsulation on interface Serial2 failed
```

Related Commands

Command	Description
cdp tlv	Configures location support in CDP.
show cdp tlv	Displays information about CDP TLVs.

debug cdp ip

To enable debug output for the IP routing information that is carried and processed by the Cisco Discovery Protocol (CDP), use the **debugcdpip**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cdp ip

no debug cdp ip

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

CDP is a media- and protocol-independent device-discovery protocol that runs on all Cisco routers.

You can use the **debugcdpip**command to determine the IP network prefixes CDP is advertising and whether CDP is correctly receiving this information from neighboring routers.

Use the **debugcdpip**command with the **debugiprouting** command to debug problems that occur when on-demand routing (ODR) routes are not installed in the routing table at a hub router. You can also use the **debugcdpip**command with the **debugcdppacket**and**debugcdpadjacency** commands along with encapsulation-specific debug commands to debug problems that occur in the receipt of CDP IP information.

Examples

The following is sample output from the**debugcdpip** command. This example shows the transmission of IP-specific information in a CDP update. In this case, three network prefixes are being sent, each with a different network mask.

```
Router# debug cdp ip
CDP-IP: Writing prefix 172.1.69.232.112/28
CDP-IP: Writing prefix 172.19.89.0/24
CDP-IP: Writing prefix 11.0.0.0/8
```

In addition to these messages, you might see the following messages:

- This message indicates that CDP is attempting to install the prefix 172.16.1.0/24 into the IP routing table:

```
CDP-IP: Updating prefix 172.16.1.0/24 in routing table
```

- This message indicates a protocol error occurred during an attempt to decode an incoming CDP packet:

```
CDP-IP: IP TLV length (3) invalid
```

- This message indicates the receipt of the IP prefix 172.16.1.0/24 from a CDP neighbor connected via Ethernet interface 0/0. The neighbor IP address is 10.0.01.

```
CDP-IP: Reading prefix 172.16.1.0/24 source 10.0.0.1 via Ethernet0/0
```

Related Commands

Command	Description
debug ip routing	Displays information on RIP routing table updates and route cache updates.

debug cef

To enable the display of information about Cisco Express Forwarding events, use the **debugcef**command in privileged EXEC mode. To disable the display of Cisco Express Forwarding events, use the **no** form of this command.

debug cef {**all**| **assert**| **background**| **broker**| **consistency-check**| **elog**| **epoch**| **fib** [**attached export**| **subblock**]| **hardware** {**notification**| **queries**}| **hash**| **high-availability**| **interest**| **interface**| **iprm**| **issu**| **loadinfo**| **memory**| **non-ip**| **path** [**extension**| **list**| **scope**]| **subtree context**| **switching background**| **table**| **xdr**}

no debug cef {**all**| **assert**| **background**| **broker**| **consistency-check**| **elog**| **epoch**| **fib** [**attached export**| **subblock**]| **hardware** {**notification**| **queries**}| **hash**| **high-availability**| **interest**| **interface**| **iprm**| **issu**| **loadinfo**| **memory**| **non-ip**| **path** [**extension**| **list**| **scope**]| **subtree context**| **switching background**| **table**| **xdr**}

Syntax Description

all	Displays debug messages for all Cisco Express Forwarding events.
assert	Displays debug messages for Cisco Express Forwarding assert events.
background	Displays debug messages for Cisco Express Forwarding background events.
broker	Displays debug messages for Cisco Express Forwarding broker events.
consistency-check	Displays debug messages for Cisco Express Forwarding consistency checker events.
elog	Displays debug messages for Cisco Express Forwarding elog events.
epoch	Displays debug messages for Cisco Express Forwarding epoch events.
fib [**attachedexport** | **subblock**]	Displays debug messages for Cisco Express Forwarding Forwarding Information Base entry events.
hardware {**notification** | **queries**}	Displays debug messages for Cisco Express Forwarding hardware API notifications or hardware API queries.
hash	Displays debug messages for Cisco Express Forwarding load-balancing hash algorithms.

high-availability	Displays debug messages for Cisco Express Forwarding high availability events.
interest	Displays debug messages for Cisco Express Forwarding interest list events.
interface	Displays debug messages for Cisco Express Forwarding interface events.
iprm	Displays debug messages for Cisco Express Forwarding IP rewrite manager events. (This keyword is not available in Cisco IOS Release 12.2(33)SRA.)
issu	Displays debug messages for Cisco Express Forwarding In Service Software Upgrade (ISSU) events.
loadinfo	Displays debug messages for Cisco Express Forwarding loadinfo events.
memory	Displays debug messages for Cisco Express Forwarding memory events.
non-ip	Displays debug messages for Cisco Express Forwarding non-IP entry events.
path [**extension** | **list** | **scope**]	Displays debug messages for Cisco Express Forwarding path events.
subtree context	Displays debug messages for Cisco Express Forwarding subtree context events.
switching background	Displays debug messages for Cisco Express Forwarding switching background events.
table	Displays debug messages for Cisco Express Forwarding table events.
xdr	Displays debug messages for Cisco Express Forwarding External Data Representation (XDR) events.

Command Default Debugging information about Cisco Express Forwarding events is not displayed.

Command Modes Privileged EXEC (#)

Command History

Release	Modification
12.2(25)S	This command was introduced. The **debugceffibattachedexport**command rerplaces the **debugipcefadjfib** command.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, you should use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, you should use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Examples

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

```
Router# debug cef all
06:23:38: HW-API: Counter poll: Label[label=implicit-null]
06:23:38: HW-API: Counter poll: Label[label=implicit-null]
06:23:38: HW-API: Counter poll: Label[label=implicit-null]
06:23:43: FIBbg: Timer 'FIB checkers: IPv4 scan-rib-ios scanner' expired, callin
g 0x40FA03FC, context 0x00010003)
06:23:43: FIBbg: Restarting timer 'FIB checkers: IPv4 scan-rib-ios scanner' with
 delay 60000
06:23:43: FIBbg: Timer 'FIB checkers: IPv4 scan-ios-rib scanner' expired, callin
g 0x40FA03FC, context 0x00010004)
06:23:43: FIBbg: Restarting timer 'FIB checkers: IPv4 scan-ios-rib scanner' with
 delay 60000
06:23:43: FIBbg: Timer 'FIB checkers: IPv6 scan-ios-rib scanner' expired, callin
g 0x40FA03FC, context 0x00020004)
06:23:43: FIBbg: Restarting timer 'FIB checkers: IPv6 scan-ios-rib scanner' with
 delay 60000
06:23:43: FIBbg: Timer 'FIB checkers: IPv4 scan-rp-lc scanner' expired, calling
0x40FA03FC, context 0x00010002)
06:23:43: FIBbg: Restarting timer 'FIB checkers: IPv4 scan-rp-lc scanner' with d
elay 60000
06:23:43: FIBbg: Timer 'FIB checkers: IPv6 scan-rp-lc scanner' expired, calling
0x40FA03FC, context 0x00020002)
06:23:43: FIBbg: Restarting timer 'FIB checkers: IPv6 scan-rp-lc scanner' with d
elay 60000
06:23:48: HW-API: Counter poll: Label[label=implicit-null]
06:23:48: HW-API: Counter poll: Label[label=implicit-null]
06:23:48: HW-API: Counter poll: Label[label=implicit-null]
06:23:58: HW-API: Counter poll: Label[label=implicit-null]
06:24:06: FIBtable: IPv4: Event modified, 0.0.0.0/0, vrf Default, 1 path, flags
00420005
06:24:06: FIBpath: Configuring IPv4 path 444B2AB0 from rib (idb=NULL, gw=9.1.41.
1, gw_table=0, rr=1) and host prefix 0.0.0.0
```

```
06:24:06: FIBpath: Configured recursive-nexthop 9.1.41.1[0] 444B2AB0 path
06:24:06: FIBfib: [v4-0.0.0.0/0 (44AAC750)] Mod type - null
06:24:06: FIBtable: IPv4: Event up, default, 0.0.0.0/0, vrf Default, 1 path, fla
gs 00420005
06:24:06: FIBtable: IPv4: Adding route for 0.0.0.0/0 but route already exists. T
rying modify.
06:24:06: FIBpath: Configuring IPv4 path 444B2AA0 from rib (idb=NULL, gw=9.1.41.
1, gw_table=0, rr=1) and host prefix 0.0.0.0sh ip
06:24:06: FIBpath: Configured recursive-nexthop 9.1.41.1[0] 444B2AA0 path
06:24:06: FIBfib: [v4-0.0.0.0/0 (44AAC750)] Mod type - null vrf
06:24:07: FIBbg: Timer 'FIB checkers: IPv4 scan-hw-sw scanner' expired, calling
0x40FA03FC, context 0x00010005)
06:24:07: FIBbg: Restarting timer 'FIB checkers: IPv4 scan-hw-sw scanner' with d
elay 60000
06:24:07: FIBbg: Timer 'FIB checkers: IPv4 scan-sw-hw scanner' expired, calling
0x40FA03FC, context 0x00010006)
06:24:07: FIBbg: Restarting timer 'FIB checkers: IPv4 scan-sw-hw scanner' with d
elay 60000
  Name                             Default RD          Interfaces
  red                              1:1                 Ethernet4/0/5
```

Related Commands

Command	Description
cef table consistency-check	Enables Cisco Express Forwarding consistency checker table values by type and parameter.
clear cef table	Clears the Cisco Express Forwarding tables.
clear ip cef inconsistency	Clears Cisco Express Forwarding inconsistency statistics and records found by the Cisco Express Forwarding consistency checkers.
debug ip cef table	Enables the collection of events that affect entries in the Cisco Express Forwarding tables.
show cef table consistency-check	Displays Cisco Express Forwarding consistency checker table values.
show ip cef inconsistency	Displays Cisco Express Forwarding IP prefix inconsistencies.

debug cell-hwic driver

To debug the Cisco IOS driver for the cellular interface, use the **debugcell-hwicdriver**command in EXEC mode.

debug cell-hwic *slotwic_slotport* **driver** {**crcdump**| **errdump**| **errors**}

Syntax Description

slot/wic_slot/port	Numeric values that indicate the router slot, WAN interface card (WIC) slot, and port.
crcdump	CRC error details.
errdump	Other error details.
errors	Errors debugging.

Command Default

None

Command Modes

EXEC (#)

Command History

Release	Modification
12.4(11)XV	This command was introduced.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)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 for debugging purposes only.

Related Commands

Command	Description
debug cellular messages async	Debugs cellular async.
debug cellular messages data	Prints Cisco IOS data path debug messages.
debug cellular firmware	Displays Cisco IOS firmware information.

Command	Description
debug cellular messages management	Prints management path messages, such as CnS.
debug cellular messages dm	Prints diagnostics monitor (DM) messages from the Qualcomm CDMA chipset.
debug cell-hwic virt-con	Redirects the Nios II console driver messages to display them in the Cisco IOS router console environment.

debug cell-hwic firmware

To see the Cisco IOS firmware information, use the **debugcell-hwicfirmware**command in EXEC mode.

debug cellular *slotwic_slotport* **firmware**

Syntax Description

slot/wic_slot/port	Numeric values that indicate the router slot, WAN interface card (WIC) slot, and port.

Command Default

None

Command Modes

EXEC

Command History

Release	Modification
12.4(11)XV	This command was introduced.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.
12.4(22)YB1	This command was integrated into Cisco IOS Release 12.4(22)YB1.

Usage Guidelines

Use this command for debugging purposes only.

Related Commands

Command	Description
debug cellular messages async	Debugs cellular async.
debug cellular messages data	Prints Cisco IOS data path debug messages.
debug cellular firmware	Debugs the Cisco IOS driver.
debug cellular messages management	Prints management path messages, such as CnS.
debug cellular messages dm	Prints diagnostics monitor (DM) messages from the Qualcomm CDMA chipset.
debug cell-hwic virt-con	Redirects the Nios II console driver messages to display them in the Cisco IOS router console environment.

debug cellular messages all

To print all Cisco IOS driver debug messages, use the **debugcellularmessagesall**command in EXEC mode.

debug cellular *slotwic_slotport* **messages all**

Syntax Description

slot/wic_slot/port	Numeric values that indicate the router slot, WAN interface card (WIC) slot, and port.

Command Default

None

Command Modes

EXEC

Command History

Release	Modification
12.4(11)XV	This command was introduced.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.

Usage Guidelines

Use this command for debugging purposes only.

Related Commands

Command	Description
debug cellular messages async	Debugs cellular async.
debug cellular messages data	Prints Cisco IOS data path debug messages.
debug cell-hwic driver	Debugs the Cisco IOS driver.
debug cell-hwic firmware	Displays Cisco IOS firmware information.
debug cellular messages management	Prints management path messages, such as CnS.
debug cellular messages dm	Prints diagnostics monitor (DM) messages from the Qualcomm CDMA chipset.
debug cellular messages virt-con	Redirects the Nios II console driver messages to display them in the Cisco IOS router console environment.

debug cellular messages async

To debug cellular async, use the **debugcellularmessagesasync**command in EXEC mode.

debug cellular *slotwic_slotport* **messages async**

Syntax Description

slot/wic_slot/port	Numeric values that indicate the router slot, WAN interface card (WIC) slot, and port.

Command Default

None

Command Modes

EXEC (#)

Command History

Release	Modification
12.4(11)XV	This command was introduced.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)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 for debugging purposes only.

Related Commands

Command	Description
debug cellular messages all	Prints all Cisco IOS driver debug messages.
debug cellular messages data	Prints Cisco IOS data path debug messages.
debug cellular driver	Debugs the Cisco IOS driver.
debug cellular firmware	Displays Cisco IOS firmware information.
debug cellular messages management	Prints management path messages, such as CnS.
debug cellular messages dm	Prints diagnostics monitor (DM) messages from the Qualcomm CDMA chipset.

Command	Description
debug cellular messages virt-con	Redirects the Nios II console driver messages to display them in the Cisco IOS router console environment.

debug cellular messages data

To print Cisco IOS data path debug messages, use the **debugcellularmessagesdata**command in EXEC mode.

show cellular *slotwic_slotport* **messages data**

Syntax Description

slot/wic_slot/port	Numeric values that indicate the router slot, WAN interface card (WIC) slot, and port.

Command Default

None

Command Modes

EXEC

Command History

Release	Modification
12.4(11)XV	This command was introduced.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.

Usage Guidelines

Use this command for debugging purposes only.

Related Commands

Command	Description
debug cellular messages all	Prints all Cisco IOS driver debug messages.
debug cellular messages async	Debugs cellular async.
debug cell-hwic driver	Debugs the Cisco IOS driver.
debug cell-hwic firmware	Displays Cisco IOS firmware information.
debug cellular messages management	Prints management path messages, such as CnS.
debug cellular messages dm	Prints diagnostics monitor (DM) messages from the Qualcomm CDMA chipset.
debug cellular messages virt-con	Redirects the Nios II console driver messages to display them in the Cisco IOS router console environment.

debug cellular messages dm

To print Diagnostics Monitor (DM) messages from the Qualcomm CDMA chipset, use the **debugcellularmessagesdm**command in EXEC mode.

debug cellular *slotwic_slotport* **messages dm**

Syntax Description

slot/wic_slot/port	Numeric values that indicate the router slot, WAN interface card (WIC) slot, and port.

Command Default

There is no default for this command.

Command Modes

EXEC

Command History

Release	Modification
12.4(11)XV	This command was introduced.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.

Usage Guidelines

Use this command for debugging purposes only.

Related Commands

Command	Description
debug cellular messages all	Prints all Cisco IOS driver debug messages.
debug cellular messages async	Debugs cellular async.
debug cellular messages data	Prints Cisco IOS data path debug messages.
debug cell-hwic driver	Debugs the Cisco IOS driver.
debug cell-hwic firmware	Displays Cisco IOS firmware information.
debug cellular messages management	Prints management path messages, such as CnS.
debug cellular messages virt-con	Redirects the Nios II console driver messages to display them in the Cisco IOS router console environment.

debug cellular messages management

To print management path messages, such as CnS, use the **debugcellularmessagesmanagement**command in EXEC mode.

debug cellular *slotwic_slotport* **messages management**

Syntax Description

slot/wic_slot/port	Numeric values that indicate the router slot, WAN interface card (WIC) slot, and port.

Command Default

None

Command Modes

EXEC

Command History

Release	Modification
12.4(11)XV	This command was introduced.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.

Usage Guidelines

Use this command for debugging purposes only.

Related Commands

Command	Description
debug cellular messages all	Prints all Cisco IOS driver debug messages.
debug cellular messages async	Debugs cellular async.
debug cellular messages data	Prints Cisco IOS data path debug messages.
debug cell-hwic driver	Debugs the Cisco IOS driver.
debug cell-hwic firmware	Displays Cisco IOS firmware information.
debug cellular messages virt-con	Redirects the Nios II console driver messages to display them in the Cisco IOS router console environment.

debug cell-hwic virt-con

To redirect the Nios II console driver messages to display them in the Cisco IOS router console environment, use the **debugcell-hwicvirt-con**command in EXEC mode.

debug cell-hwic *slotwic_slotport* **virt-con** {**clear**| **disable**| **dump-data-structs**| **log**| **monitor**| **wrapper-on**| **wrapper-off**}

Syntax Description

slot/wic_slot/port	Numeric values that indicate the router slot, WAN interface card (WIC) slot, and port.
clear	(Optional) Clears all virtual console debug log messages.
disable	(Optional) Disables virtual console real-time debug monitoring.
dump-data-structurs	(Optional) Dumps virtual console data structures.
log	(Optional) Displays virtual console messages from the debug log.
monitor	(Optional) Enables monitoring of real-time virtual console debug messages.
wrapper-on	(Optional) Disables wraparound for virtual console log messages.
wrapper-off	(Optional) Enables wraparound for virtual console log messages.

Command Default

There is no default for this command.

Command Modes

EXEC (#)

Command History

Release	Modification
12.4(11)XV	This command was introduced.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)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.

Release	Modification
12.4(22)YB1	This command was integrated into Cisco IOS Release 12.4(22)YB1.

Usage Guidelines

Use this command for debugging purposes only.

Related Commands

Command	Description
debug cellular messages all	Prints all Cisco IOS driver debug messages.
debug cellular messages async	Debugs cellular async.
debug cellular messages data	Prints Cisco IOS data path debug messages.
debug cell-hwic driver	Debugs the Cisco IOS driver.
debug cell-hwic firmware	Displays Cisco IOS firmware information.
debug cellular messages management	Prints management path messages, such as CnS.
debug cellular messages dm	Prints diagnostics monitor (DM) messages from the Qualcomm CDMA chipset.

debug cem ls errors

To debug connection errors or null data structures, use the debug cem ls errors command in privileged EXEC mode. To disable this form of debugging, use the no form of this command.

debug cem ls errors

no *debug cem ls errors*

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRC	This command was introduced on the Cisco 7600 series routers.

Usage Guidelines

Use the show debug command to see debug information.

Examples

The following command turns on CEM local switching error debugging:

```
Router# debug cem ls errors
```

Related Commands

Command	Description
debug cem ls events	Enables debugging of events relating to CEM local switching.

debug cem ls events

To debug CEM local switching events, use the debug cem ls events command in privileged EXEC mode. To disable this form of debugging, use the no form of this command.

debug cem ls events

no *debug cem ls events*

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRC	This command was introduced on the Cisco 7600 series routers.

Usage Guidelines

Use the show debug command to see debug information.

Examples

The following command turns on debugging for CEM local switching events.

```
Router# debug cem ls events
```

Related Commands

Command	Description
debug cem ls errors	Enables debugging of connection errors or null data structures.

debug ces-conn

To display information from circuit emulation service (CES) clients, use the **debugces-conn**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ces-conn [**all**| **errors**| **events**]

no debug ces-conn

Syntax Description

all	(Optional) Displays all error and event information.
errors	(Optional) Displays only error information.
events	(Optional) Displays only event information.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)XM	This command is supported on Cisco 3600 series routers.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.

Examples

The following example shows debug output for a CES connection:

```
Router# debug ces-conn all
CES all debugging is on
Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# connect conn1 t1 3/0 1 atm1/0 1/100
Router(config-ces-conn)# exit
Router(config)#
*Mar  6 18:32:27:CES_CLIENT:vc QoS parameters are PCR = 590, CDV =
5000, CAS_ENABLED = 1,partial fill = 0, multiplier = 8,cbr rate = 64,
clock recovery = 0,service_type = 3, error method = 0,sdt_size = 196,
billing count = 0
*Mar  6 18:32:27:CES_CLIENT:attempt 1 to activate segment>
```

debug cfm

To enable debugging of the data path of Ethernet connectivity fault management (CFM) on Cisco Catalyst 6500 series switches, use the **debug cfm** command in privileged EXEC mode. To disable the debugging function, use the **no** form of this command.

debug cfm {**all**| **api**| **cfmpal**| **common**| **db**| **isr**}

no debug cfm {**all**| **api**| **cfmpal**| **common**| **db**| **isr**}

Syntax Description

all	Specifies all Catalyst 6500 switch-specific route processor and switch processor (RP/SP) events.
api	Specifies Catalyst 6500 switch-specific application program interface (API) events.
cfmpal	Specifies general Catalyst 6500 switch debugging.
common	Specifies common Catalyst 6500 switch RP/SP components.
db	Specifies Catalyst 6500 switch CFM database debugging.
isr	Specifies Catalyst 6500 switch-specific ingress CFM packet debugging.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SXI2	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

The output from this command is a log of activity.

Use this command to troubleshoot Ethernet CFM on Cisco Catalyst 6500 series switches.

Examples

The following example shows output of the **debug cfm all** command:

```
Device# debug cfm all

CFM DB events debugging is on
CFM Ingress ISR events debugging is on
CFMPAL events debugging is on
CFM API events debugging is on
CFM RP/SP COMMON events debugging is on
CFM packets debugging is on
```

debug channel events

To display processing events on Cisco 7000 series routers that occur on the channel adapter interfaces of all installed adapters, use the **debugchannelevents** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug channel events

no debug channel 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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

This command displays CMCC adapter events that occur on the Channel Interface Processor (CIP) or Channel Port Adapter (CPA) and is useful for diagnosing problems in an IBM channel attach network. It provides an overall picture of the stability of the network. In a stable network, the **debugchannelevents** command does not return any information. If the command generates numerous messages, the messages can indicate the possible source of the problems. To observe the statistic message (cip_love_letter) sent every 10 seconds, use the **debugchannellove** command.

When configuring or making changes to a router or interface that supports IBM channel attach, enable the **debugchannelevents** 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 sample output is from the **debugchannelevents** command:

```
Router# debug channel events
Channel3/0: cip_reset(), state administratively down
Channel3/0: cip_reset(), state up
Channel3/0: sending nodeid
Channel3/0: sending command for vc 0, CLAW path C700, device C0
```

The following line indicates that the CIP is being reset to an administrative down state:

```
Channel3/0: cip_reset(), state administratively down
```

The following line indicates that the CIP is being reset to an administrative up state:

```
Channel3/0: cip_reset(), state up
```

The following line indicates that the node ID is being sent to the CIP. This information is the same as the "Local Node" information under the **showextendedchannel***slot*/*port***subchannels** command. The CIP needs to send this information to the host mainframe.

```
Channel3/0: sending nodeid
```

The following line indicates that a Common Link Access for Workstations (CLAW) subchannel command is being sent from the Route Processor (RP) to the CIP. The value vc 0 indicates that the CIP will use virtual circuit number 0 with this device. The virtual circuit number also shows up when you use the **debugchannelpackets**command.

```
Channel3/0: sending command for vc 0, CLAW path C700, device C0
```

The following is a sample output that is generated by the **debugchannelevents**command when a CMPC+ IP TG connection is activated with the host:

```
1d05h:Channel4/2:Received route UP  for tg (768)
1d05h:Adding STATIC ROUTE for vc:768
```

The following is a sample output from the **debugchannelevents**command when a CMPC+ IP TG connection is deactivated:

```
1d05h:Channel4/2:Received route DOWN  for tg (768)
1d05h:Deleting STATIC ROUTE for vc:768
```

Related Commands

Command	Description
debug channel ilan	Displays CIP love letter events.
debug channel packets	Displays per-packet debugging output.

debug channel ilan

To display messages relating to configuration and bridging using Cisco Mainframe Channel Connection (CMCC) internal LANs and to help debug source-route bridging (SRB) problems related to CMCC internal LANs, use the **debugchannelilan** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug channel ilan

no debug channel ilan

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.0(3)	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The debug channel ilan command displays events related to CMCC internal LANs. This command is useful for debugging problems associated with CMCC internal LAN configuration. It is also useful for debugging problems related to SRB packet flows through internal LANs.

Examples

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

```
Router# debug channel ilan
Channel internal LANs debugging is on
```

The following line indicates that a packet destined for the CMCC via a configured internal MAC adapter configured on an internal LAN was dropped because the Logical Link Control (LLC) end station in Cisco IOS software did not exist:

```
CIP ILAN(Channel3/2-Token): Packet dropped - NULL LLC
```

The following line indicates that a packet destined for the CMCC via a configured internal MAC adapter configured on an internal LAN was dropped because the CMCC had not yet acknowledged the internal MAC adapter configuration command:

```
Channel3/2: ILAN Token-Ring 3 - CIP internal MAC adapter not acknowledged DMAC(4000.7000.0001)
 SMAC(0c00.8123.0023)
```

Related Commands

Command	Description
debug channel events	Displays processing that occurs on the channel adapter interfaces of all installed adapters.
debug source bridge	Displays information about packets and frames transferred across a source-route bridge.

debug channel love

To display Channel Interface Processor (CIP) love letter events, use the **debugchannellove** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug channel love

no debug channel love

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command displays CIP love letter events (an operating status or configuration message) that occur on the CIP interface processor and is useful for diagnosing problems in an IBM channel attach network. It provides an overall picture of the stability of the network. In a stable network, the **debugchannellove** command returns a statistic message (cip_love_letter) that is sent every 10 seconds. This command is valid for the Cisco 7000 series routers only.

Examples

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

```
Router# debug channel love
Channel3/1: love letter received, bytes 3308
Channel3/0: love letter received, bytes 3336
cip_love_letter: received ll, but no cip_info
```

The following line indicates that data was received on the CIP:

```
Channel3/1: love letter received, bytes 3308
```

The following line indicates that the interface is enabled, but there is no configuration for it. It does not normally indicate a problem, just that the Route Processor (RP) got statistics from the CIP but has no place to store them.

```
cip_love_letter: received ll, but no cip_info
```

Related Commands

Command	Description
debug channel events	Displays processing that occurs on the channel adapter interfaces of all installed adapters.
debug channel packets	Displays per-packet debugging output.

debug channel packets

To display per-packet debugging output, use the **debugchannelpackets** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug channel packets

no debug channel packets

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The **debugchannelpackets** command displays all process-level Channel Interface Processor (CIP) packets for both outbound and inbound packets. The output reports information when a packet is received or a transmission is attempted. You will need to disable fast switching and autonomous switching to obtain debugging output. This command is useful for determining whether packets are received or sent correctly.

This command is valid for the Cisco 7000 series routers only.

Examples

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

```
Router# debug channel packets

(Channel3/0)-out size = 104, vc = 0000, type = 0800, src 172.24.0.11, dst 172.24.1.58
(Channel3/0)-in size = 48, vc = 0000, type = 0800, src 172.24.1.58, dst 172.24.15.197
(Channel3/0)-in size = 48, vc = 0000, type = 0800, src 172.24.1.58, dst 172.24.15.197
(Channel3/0)-out size = 71, vc = 0000, type = 0800, src 172.24.15.197, dst 172.24.1.58
(Channel3/0)-in size = 44, vc = 0000, type = 0800, src 172.24.1.58, dst 172.24.15.197
```

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

Table 20: debug channel packets Field Descriptions

Field	Description
(Channel3/0)	Interface slot and port.
in/out	"In" is a packet from the mainframe to the router. "Out" is a packet from the router to the mainframe.
size =	Number of bytes in the packet, including internal overhead.
vc =	Value from 0 to 511 that maps to the **claw** interface configuration command. This information is from the MAC layer.

Field	Description
type =	Encapsulation type in the MAC layer. The value 0800 indicates an IP datagram.
src	Origin, or source, of the packet, as opposed to the previous hop address.
dst	Destination of the packet, as opposed to the next-hop address.