The following command has been removed from documentation in Cisco IOS Release 12.2(4)B and will not appear in future releases of the Cisco IOS software documentation set. The command has been replaced by the **debug ssg tcp-redirect** command.

- **debug ssg http-redirect**

The following commands have been removed from documentation in Cisco IOS Software Release 12.2(11)T and will not appear in future releases of the Cisco IOS software documentation set. The commands are replaced by the **debug ss7 mtp2** command.

- **debug ss7 mtp2 aerm**
- **debug ss7 mtp2 backaul**
- **debug ss7 mtp2 col1ng**
- **debug ss7 mtp2 iac**
- **debug ss7 mtp2 lsc**
- **debug ss7 mtp2 lssu**
- **debug ss7 mtp2 msu**
- **debug ss7 mtp2 packet**
- **debug ss7 mtp2 rcv**
- **debug ss7 mtp2 seurm**
- **debug ss7 mtp2 timer**
- **debug ss7 mtp2 txc**

debug source event

To display information on source-route bridging activity, use the **debug source event** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug source event

no debug source event

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Some of the output from the **debug source bridge** and **debug source error** commands is identical to the output of this command.

Note In order to use the **debug source event** command to display traffic source-routed through an interface, you first must disable fast switching of SRB frames with the **no source bridge route-cache** interface configuration command.

Examples

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

```
Router# debug source event

RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9
[0800.3201.00A1.0050]
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9
[0800.3201.00A1.0050]
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9
[0800.3201.00A1.0050]
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9
[0800.3201.00A1.0050]
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9
[0800.3201.00A1.0050]
```

Table 230 describes the significant fields shown in the display.

Table 230 debug source event Field Descriptions

Field	Description
RSRB0:	Indication that this RIF cache entry is for the Token Ring interface 0, which has been configured for remote source-route bridging (SRB). (SRB1, in contrast, would indicate that this RIF cache entry is for Token Ring 1, configured for SRB.)
forward	Forward (normal data) packet, in contrast to a control packet containing proprietary Cisco bridging information.
srn 5	Ring number of the source ring of the packet.

Table 230 debug source event Field Descriptions (continued)

Field	Description
bn 1	Bridge number of the bridge this packet traverses.
trn 10	Ring number of the target ring of the packet.
src: 8110.2222.33c1	Source address of the route in this RIF cache entry.
dst: 1000.5a59.04f9	Destination address of the route in this RIF cache entry.
[0800.3201.00A1.0050]	RIF string in this RIF cache entry.

In the following example messages, SRBnumber or RSRBnumber denotes a message associated with interface Token Ring number. A number of 99 denotes the remote side of the network.

```
SRBnumber: no path, s: source-MAC-addr d: dst-MAC-addr rif: rif
```

In the preceding example, a bridgeable packet came in on interface Token Ring number but there was nowhere to send it. This is most likely a configuration error. For example, an interface has source bridging turned on, but it is not connected to another source bridging interface or a ring group.

In the following example, a bridgeable packet has been forwarded from Token Ring number to the target ring. The two interfaces are directly linked.

```
SRBnumber: direct forward (srn ring bn bridge trn ring)
```

In the following examples, a proxy explorer reply was not generated because the address could not be reached from this interface. The packet came from the node with the first address.

```
SRBnumber: br dropped proxy XID, address for address, wrong vring (rem)
SRBnumber: br dropped proxy TEST, address for address, wrong vring (rem)
SRBnumber: br dropped proxy XID, address for address, wrong vring (local)
SRBnumber: br dropped proxy TEST, address for address, wrong vring (local)
SRBnumber: br dropped proxy XID, address for address, no path
SRBnumber: br dropped proxy TEST, address for address, no path
```

In the following example, an appropriate proxy explorer reply was generated on behalf of the second address. It is sent to the first address.

```
SRBnumber: br sent proxy XID, address for address[rif]
SRBnumber: br sent proxy TEST, address for address[rif]
```

The following example indicates that the broadcast bits were not set, or that the routing information indicator on the packet was not set:

```
SRBnumber: illegal explorer, s: source-MAC-addr d: dst-MAC-addr rif: rif
```

The following example indicates that the direction bit in the RIF field was set, or that an odd packet length was encountered. Such packets are dropped.

```
SRBnumber: bad explorer control, D set or odd
```

The following example indicates that a spanning explorer was dropped because the spanning option was not configured on the interface:

```
SRBnumber: span dropped, input off, s: source-MAC-addr d: dst-MAC-addr rif: rif
```

The following example indicates that a spanning explorer was dropped because it had traversed the ring previously:

```
SRBnumber: span violation, s: source-MAC-addr d: dst-MAC-addr rif: rif
```

The following example indicates that an explorer was dropped because the maximum hop count limit was reached on that interface:

```
SRBnumber: max hops reached - hop-cnt, s: source-MAC-addr d: dst-MAC-addr rif: rif
```

The following example indicates that the ring exchange request was sent to the indicated peer. This request tells the remote side which rings this node has and asks for a reply indicating which rings that side has.

```
RSRB: sent RingXreq to ring-group/ip-addr
```

The following example indicates that a message was sent to the remote peer. The label variable can be AHDR (active header), PHDR (passive header), HDR (normal header), or DATA (data exchange), and op can be Forward, Explorer, Ring Xchg, Req, Ring Xchg, Rep, Unknown Ring Group, Unknown Peer, or Unknown Target Ring.

```
RSRB: label: sent op to ring-group/ip-addr
```

The following example indicates that the remote bridge and ring pair were removed from or added to the local ring group table because the remote peer changed:

```
RSRB: removing bn bridge rn ring from ring-group/ip-addr
RSRB: added bridge bridge, ring ring for ring-group/ip-addr
```

The following example shows miscellaneous remote peer connection establishment messages:

```
RSRB: peer ring-group/ip-addr closed [last state n]
RSRB: passive open ip-addr(remote port) -> local port
RSRB: CONN: opening peer ring-group/ip-addr, attempt n
RSRB: CONN: Remote closed ring-group/ip-addr on open
RSRB: CONN: peer ring-group/ip-addr open failed, reason[code]
```

The following example shows that an explorer packet was propagated onto the local ring from the remote ring group:

```
RSRBn: sent local explorer, bridge bridge trn ring, [rif]
```

The following messages indicate that the RSRB code found that the packet was in error:

```
RSRBn: ring group ring-group not found
RSRBn: explorer rif [rif] not long enough
```

The following example indicates that a buffer could not be obtained for a ring exchange packet (this is an internal error):

```
RSRB: couldn't get pak for ringXchg
```

The following example indicates that a ring exchange packet was received that had an incorrect length (this is an internal error):

```
RSRB: XCHG: req/reply badly formed, length pak-length, peer peer-id
```

The following example indicates that a ring entry was removed for the peer; the ring was possibly disconnected from the network, causing the remote router to send an update to all its peers.

```
RSRB: removing bridge bridge ring ring from peer-id ring-type
```

The following example indicates that a ring entry was added for the specified peer; the ring was possibly added to the network, causing the other router to send an update to all its peers.

```
RSRB: added bridge bridge, ring ring for peer-id
```

The following example indicates that no memory was available to add a ring number to the ring group specified (this is an internal error):

```
RSRB: no memory for ring element ring-group
```

The following example indicates that memory was corrupted for a connection block (this is an internal error):

```
RSRB: CONN: corrupt connection block
```

The following example indicates that a connector process started, but that there was no packet to process (this is an internal error):

```
RSRB: CONN: warning, no initial packet, peer: ip-addr peer-pointer
```

The following example indicates that a packet was received with a version number different from the one pre-sent on the router:

```
RSRB: IF New version. local=local-version, remote=remote-version,pak-op-code peer-id
```

The following example indicates that a packet with a bad op code was received for a direct encapsulation peer (this is an internal error):

```
RSRB: IFin: bad op op-code (op code string) from peer-id
```

The following example indicates that the virtual ring header will not fit on the packet to be sent to the peer (this is an internal error):

```
RSRB: vrif_sender, hdr won't fit
```

The following example indicates that the specified peer is being opened. The retry count specifies the number of times the opening operation is attempted.

```
RSRB: CONN: opening peer peer-id retry-count
```

The following example indicates that the router, configured for FST encapsulation, received a version reply to the version request packet it had sent previously:

```
RSRB: FST Rcvd version reply from peer-id (version version-number)
```

The following example indicates that the router, configured for FST encapsulation, sent a version request packet to the specified peer:

```
RSRB: FST Version Request. op = opcode, peer-id
```

The following example indicates that the router received a packet with a bad op code from the specified peer (this is an internal error):

```
RSRB: FSTin: bad op opcode (op code string) from peer-id
```

The following example indicates that the TCP connection between the router and the specified peer is being aborted:

```
RSRB: aborting ring-group/peer-id (vrtcpd_abort called)
```

The following example indicates that an attempt to establish a TCP connection to a remote peer timed out:

```
RSRB: CONN: attempt timed out
```

The following example indicates that a packet was dropped because the ring group number in the packet did not correlate with the ring groups configured on the router:

```
RSRBnumber: ring group ring-group not found
```

debug span

To display information on changes in the spanning-tree topology when debugging a transparent bridge, use the **debug span** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug span

no debug span

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command is useful for tracking and verifying that the spanning-tree protocol is operating correctly.

Examples

The following is sample output from the **debug span** command for an IEEE BPDU packet:

```
Router# debug span

ST: Ether4 0000000000000A080002A02D6700000000000A080002A02D67800100001400002000F00
```

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

```
ST: Ether4 0000000000000A080002A02D6700000000000A080002A02D67800100001400002000F00
          A   B C D E   F          G      H   I          J K L   M   N   O
```

Table 231 describes the significant fields shown in the display.

Table 231 debug span Field Descriptions—IEEE BPDU Packet

Field	Description
ST:	Indication that this is a spanning tree packet.
Ether4	Interface receiving the packet.
(A) 0000	Indication that this is an IEEE BPDU packet.
(B) 00	Version.
(C) 00	Command mode: • 00 indicates config BPDU. • 80 indicates the Topology Change Notification (TCN) BPDU.
(D) 00	Topology change acknowledgment: • 00 indicates no change. • 80 indicates a change notification.
(E) 000A	Root priority.
(F) 080002A02D67	Root ID.

Table 231 debug span Field Descriptions—IEEE BPDU Packet (continued)

Field	Description
(G) 00000000	Root path cost (0 means the sender of this BPDU packet is the root bridge).
(H) 000A	Bridge priority.
(I) 080002A02D67	Bridge ID.
(J) 80	Port priority.
(K) 01	Port Number 1.
(L) 0000	Message age in 256ths of a second (0 seconds, in this case).
(M) 1400	Maximum age in 256ths of a second (20 seconds, in this case).
(N) 0200	Hello time in 256ths of a second (2 seconds, in this case).
(O) 0F00	Forward delay in 256ths of a second (15 seconds, in this case).

The following is sample output from the **debug span** command for a DEC BPDU packet:

```
Router# debug span

ST: Ethernet4 E1190100000200000C01A2C90064008000000C0106CE0A01050F1E6A
```

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

```
E1 19 01 00 0002 00000C01A2C9 0064 0080 00000C0106CE 0A 01 05 0F 1E 6A
A  B  C  D  E    F            G    H    I            J  K  L  M  N  O
```

Table 232 describes the significant fields.

Table 232 debug span Field Descriptions for a DEC BPDU Packet

Field	Description
ST:	Indication that this is a spanning tree packet.
Ethernet4	Interface receiving the packet.
(A) E1	Indication that this is a DEC BPDU packet.
(B) 19	Indication that this is a DEC hello packet. Possible values are as follows: • 0x19—DEC Hello • 0x02—TCN
(C) 01	DEC version.
(D) 00	Flag that is a bit field with the following mapping: • 1—TCN • 2—TCN acknowledgment • 8—Use short timers
(E) 0002	Root priority.
(F) 00000C01A2C9	Root ID (MAC address).
(G) 0064	Root path cost (translated as 100 in decimal notation).

Table 232 debug span Field Descriptions for a DEC BPDU Packet (continued)

Field	Description
(H) 0080	Bridge priority.
(I) 00000C0106CE	Bridge ID.
(J) 0A	Port ID (in contrast to interface number).
(K) 01	Message age (in seconds).
(L) 05	Hello time (in seconds).
(M) 0F	Maximum age (in seconds).
(N) 1E	Forward delay (in seconds).
(O) 6A	Not applicable.

debug ss7 mtp1

Note Use this command only if told to do so by your Cisco representative.

To initiate SS7 MTP1 debugging, enter the **debug ss7 mtp1** command in global configuration mode during a low-traffic period. To disable debugging output, use the **no** form of this command.

debug ss7 mtp1 [**mtp2 | ipc | link-state | oir | rx | scc-regs | siram | tdm-info | tx**]

no debug ss7 mtp1

Syntax Description

mtp2	Initiates SS7 MTP2 debugging.
ipc	Initiates SS7 MTP1 debugging for HOST/FW IPC.
link-state	Initiates SS7 MTP1 debugging for link-state transitions.
oir	Initiates SS7 MTP1 trunk dial feature card (DFC) online insertion and removal (OIR) debugging.
rx	Initiates SS7 MTP1 debugging for receive events. Not used in this release.
scc-regs	Initiates SS7 MTP1 debugging for SCC registers. Not used in this release.
siram	Initiates SS7 MTP1 debugging for siram values. Not used in this release.
tdm-info	Initiates SS7 MTP1 debugging for time-division multiplexing (TDM) information.
tx	Initiates SS7 MTP1 debugging for transmission events. Not used in this release.

Command Modes

Global configuration

Defaults

Debug is disabled

Command History

Release	Modification
12.2(11)T	This command was introduced on the Cisco AS5350 and Cisco AS5400 Signaling Link Terminal (SLT).

Usage Guidelines

The following debug commands are not used in this release:

- **debug ss7 mtp1 rx**
- **debug ss7 mtp1 tx**
- **debug ss7 mtp1 scc-regs**
- **debug ss7 mtp1 siram**

Examples

To turn on message tracing between the host processor and the trunk firmware for each trunk card inserted, use the **debug ss7 mtp1 ipc** command.

For example, there is a digital link in slot 7, trunk 0, channel-group 0 (therefore, timeslot 1). When you enter **show ss7 mtp1 links**, the following output is displayed:

```
Router# show ss7 mtp1 links

SS7 MTP1 Links [num = 1, platform max = 4]:

                    session
interface   type   SCC     state    channel
--------- -------- --- ------------ -------
7/0:0  digital  7/3      STOPPED     0
```

Notice that the link is stopped in this example. Enter the following commands:

```
Router# debug ss7 mtp1 ipc
Router# configure terminal
Router(config)# interface serial 7/0:0
Router(config-if)# no shutdown
Router(config-if)# end
```

You would see trace output similar to the following:

```
00:01:27:from Trunk(7):TRUNK_SERIAL_STOP(3), link_type=2
00:01:27:from Trunk(7):TRUNK_SERIAL_START(3), link_type=2
```

In this case, the output means that for the SS7 link that is using SCC3 on the trunk card in slot 7 (link 7/0:0), the host processor has told the board firmware to STOP then START.

To show low-level (MTP1) state changes for the internal state-machine implemented for each SS7 link, use the **debug ss7 mtp1 link-state** command. The following output shows the different MTP1 states link Serial 7/0:0 goes through during shutdown, no shutdown, and debug.

For example, if you stopped the SS7 link 7/0:0 (shutdown), then restarted it (no shutdown), you could see MTP1 state changes by enabling debugging, as follows:

```
Router# debug ss7 mtp1 link-state
Router# configure terminal
Router(config)# interface serial 7/0:0
Router(config-if)# shutdown

01:02:20:%TRUNK_SERIAL-3-STATE_GENERIC:
At ../src-7k-as5400/as5400_ss7_link.c:511 [Serial7/0:0]:STOP:
STARTED -> STOP_PENDING
ss7_link_ll_stop 7/0:0:Tx shadow ring has
0 unsent buffers

01:02:20:%TRUNK_SERIAL-3-STATE_GENERIC:
At ../src-7k-as5400/as5400_ss7_link.c:1010 [Serial7/0:0]: FW_STOPPED:
STOP_PENDING -> STOPPED
```

Now restart the link:

```
Router(config-if)# no shutdown

01:02:26:ss7_link_start:slot=7/SCCport=3 current state is STOPPED

01:02:26:%TRUNK_SERIAL-3-STATE_GENERIC:
At ../src-7k-as5400/as5400_ss7_link.c:1417 [Serial7/0:0]: START:
STOPPED -> START_PENDING
```

```
01:02:26:%TRUNK_SERIAL-3-STATE_GENERIC:
At ../src-7k-as5400/as5400_ss7_link.c:1164 [Serial7/0:0]: STOP_START:
START_PENDING -> STOP_START_PENDING
ss7_link_ll_stop 7/0:0:Tx shadow ring has 0 unsent buffers

01:02:26:%TRUNK_SERIAL-3-STATE_GENERIC:
At ../src-7k-as5400/as5400_ss7_link.c:1010 [Serial7/0:0]: FW_STOPPED:
STOP_START_PENDING -> START_PENDING

01:02:26:%TRUNK_SERIAL-3-STATE_GENERIC:
At ../src-7k-as5400/as5400_ss7_link.c:1234 [Serial7/0:0]: FW_STARTED:
START_PENDING -> STARTED
```

To show detailed information about how TDM timeslots on the DFC trunk card on the host backplane are allocated and deallocated based on link configuration activity, use the **debug ss7 mtp1 tdm-info** command.

For example, if you wanted to create a digital SS7 link on timeslot 1 of trunk 0 for an 8PRI board in slot 7, and you would like to see traces of the TDM resources allocated, you would enable TDM debugging using the **debug ss7 mtp1 tdm-info** command then create the new SS7 link as described above, as in the following example:

```
Router# debug ss7 mtp1 tdm-info

Router# configure terminal
Router(config)# controller t1 7/0
Router(config-controller)# channel-group 0 timeslots 1
Router(config-controller)# exit
Router(config)# interface serial 7/0:0
Router(config-if)# encapsulation ss7
```

Due to the debug flag, the following information is displayed:

```
05:26:55: ss7_link_flink_tdm_setup:card type for slot 7 is T1 8PRI

05:26:55: ds0-side BEFORE call to tdm_allocate_bp_ts()
   slot    = 7
   unit    = 0     (trunk)
   channel = 4
   stream  = 0
   group   = 0

05:26:55: scc-side BEFORE call to tdm_allocate_bp_ts()
   slot    = 7
   unit    = 29
   channel = 3     (SCC-port)
   stream  = 3
   group   = 0

05:26:55:
05:26:55:TDM(PRI:0x28002000):Close PRI framer st0 ch4
05:26:55:<<<   tdm_allocate_bp_ts(ss7_ch) SUCCEEDED   >>>
05:26:55:scc-side AFTER call to tdm_allocate_bp_ts()
   bp_channel = 4
   bp_stream  = 0
   bp_ts->bp_stream    = 0
   bp_ts->bp_channel   = 4
   bp_ts->vdev_slot    = 7
   bp_ts->vdev_channel = 3
```

bp_ts->vdev_slot = 7 should be same as the CLI slot, and bp_ts->vdev_channel = 3should be *->channel.

When you later remove the SS7 link, other information is displayed showing how resources are cleaned up.

Related Commands

Command	Description
debug sse	Displays debug messages for the a Signaling System 7 (SS7) Session Manager.

debug ss7 mtp2

To trace backhaul SS7 MTP 2 message signaling units (MSUs), enter the **debug ss7 mtp2** command in global configuration mode during a low-traffic period. To disable debugging output, use the **no** form of this command.

debug ss7 mtp2 [**aerm | backhaul | cong | iac | lsc | lssu | msu | packet** [**all**] **| rcv | seurm | timer | txc**][*channel*]

no debug ss7 mtp2

Syntax Description

aerm	Initiates alignment Error Rate Monitor events.
backhaul	Initiates trace backhaul control messages. The *channel* argument represents a logical channel number. Valid values are from 0 to 3.
cong	Initiates congestion Control events.
iac	Initiates initial Alignment Control events.
lsc	Initiates Link State Control events.
lssu	Initiates trace backhaul LSSU messages.
msu	Initiates trace backhaul MSU messages (use during low traffic only).
packet [**all**]	Initiates low-level MTP2 packet tracing. If you do not specify a channel number or enter the **all** keyword, the command displays information for channel 0.
rcv	Displays information about SS7 MTP 2 receiver state machine events and transitions.
seurm	Displays information about SS7 MTP 2 Signal Unit Error Rate Monitor (SUERM) state machine events and transitions.
timer	Displays information about SS7 Message Transfer Part level 2 (MTP 2) timer starts and stops.
txc	Displays information about SS7 MTP 2 transmit state machine events and transitions.
channel	The *channel* argument represents a logical channel number. Valid values are from 0 to 3.

Command Modes

Global configuration

Defaults

Debug is disabled.

Command History

Release	Modification
12.0(7)XR	This command was introduced.
12.1(1)T	This command was integrated into Cisco IOS Release 12.1(1)T.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T. This command was implemented on the Cisco AS5350 and Cisco AS5400 Cisco Signaling Link Terminal (SLT).

Usage Guidelines

If you do not specify a channel number with each keyword, the command displays information for channel 0.

Examples

The following is an example of **debug ss7 mtp2 aerm** command output. See the MTP 2 specification tables for details:

```
Router# debug ss7 mtp2 aerm 0

*Mar  8 08:59:30.991:itu2AERM_Start  chnl=0  MTP2AERM_IDLE
*Mar  8 08:59:35.070:itu2AERM_Stop  chnl=0  MTP2AERM_MONITORING
```

The following is an example of **debug ss7 mtp2 backhaul** command output for channel 0:

```
Router# debug ss7 mtp2 backhaul 0
*Mar  1 03:08:04.433: MTP2: send Disc Ind  ch=0  reason=0x14-T2 expired waiting for SIO
*Mar  1 03:08:04.433: MTP2: send LSC Ind  ch=0  event=0x8-lost link alignment cause=0x0
*Mar  1 03:08:08.721: MTP2: rcvd Conn Req - Normal  ch=0
*Mar  1 03:08:10.311: MTP2: rcvd Statistics Req-Send&Reset   ch=0
*Mar  1 03:08:10.311: MTP2: send Stats Cfm  ch=0
*Mar  1 03:08:20.440: MTP2: send Disc Ind  ch=0  reason=0x14-T2 expired waiting for SIO
*Mar  1 03:08:20.444: MTP2: send LSC Ind  ch=0  event=0x8-lost link alignment cause=0x0
*Mar  1 03:08:24.719: MTP2: rcvd Conn Req - Normal  ch=0
*Mar  1 03:08:36.438: MTP2: send Disc Ind  ch=0  reason=0x14-T2 expired waiting for SIO
*Mar  1 03:08:36.438: MTP2: send LSC Ind  ch=0  event=0x8-lost link alignment cause=0x0
*Mar  1 03:08:40.312: MTP2: rcvd Statistics Req-Send&Reset   ch=0
*Mar  1 03:08:40.312: MTP2: send Stats Cfm  ch=0
*Mar  1 03:08:40.721: MTP2: rcvd Conn Req - Normal  ch=0
*Mar  1 03:08:52.444: MTP2: send Disc Ind  ch=0  reason=0x14-T2 expired waiting for SIO
*Mar  1 03:08:52.444: MTP2: send LSC Ind  ch=0  event=0x8-lost link alignment cause=0x0
*Mar  1 03:08:56.719: MTP2: rcvd Conn Req - Normal  ch=0
*Mar  1 03:09:08.438: MTP2: send Disc Ind  ch=0  reason=0x14-T2 expired waiting for SIO
*Mar  1 03:09:08.438: MTP2: send LSC Ind  ch=0  event=0x8-lost link alignment cause=0x0
```

The following is an example of **debug ss7 mtp2 cong** command output. See the MTP 2 specification tables for details:

```
Router# debug ss7 mtp2 cong 0

*Mar  8 09:10:56.219:itu2CongestionOnset  chnl=0  MTP2CONGESTION_IDLE
*Mar  8 09:10:59.332:itu2CongestionAbatement chnl=0
MTP2CONGESTION_ACTIVE
*Mar  8 09:11:01.143:itu2CongestionAbatement chnl=0  MTP2CONGESTION_IDLE
```

The following is an example of **debug ss7 mtp2 iac** command output. See the MTP 2 specification tables for details:

```
Router# debug ss7 mtp2 iac 0

*Mar  8 09:17:58.367:itu2IAC_Start  chnl=0  MTP2IAC_IDLE
```

```
*Mar  8 09:17:58.739:itu2IAC_Rcvd_SIO  chnl=0  MTP2IAC_NOT_ALIGNED
*Mar  8 09:17:58.739:itu2IAC_Rcvd_SIN  chnl=0  MTP2IAC_ALIGNED
*Mar  8 09:17:58.739:itu2IAC_Rcvd_SIN  chnl=0  MTP2IAC_PROVING
*Mar  8 09:18:02.814:itu2IAC_T4_TMO   chnl=0  MTP2IAC_PROVING
```

The following is an example of **debug ss7 mtp2 lsc** command output. See the MTP 2 specification tables for details:

```
Router# debug ss7 mtp2 lsc 0

*Mar  8 09:20:21.105:itu2LSC_Rcvd_SIOS  chnl=0  MTP2LSC_INSERVICE
*Mar  8 09:20:21.121:itu2LSC_Retrieve_BSNT  chnl=0  MTP2LSC_OOS
*Mar  8 09:20:22.058:itu2LSC_SetEmergency  chnl=0  MTP2LSC_OOS
*Mar  8 09:20:22.058:itu2LSC_Start  chnl=0  MTP2LSC_OOS
*Mar  8 09:20:33.785:itu2LSC_AlignmentNotPossible  chnl=0
MTP2LSC_INITIAL_ALIGNMENT
*Mar  8 09:20:38.758:itu2LSC_SetEmergency  chnl=0  MTP2LSC_OOS
*Mar  8 09:20:38.758:itu2LSC_Start  chnl=0  MTP2LSC_OOS
*Mar  8 09:20:44.315:itu2LSC_Rcvd_SIO  chnl=0  MTP2LSC_INITIAL_ALIGNMENT
*Mar  8 09:20:44.315:itu2LSC_Rcvd_SIO  chnl=0  MTP2LSC_INITIAL_ALIGNMENT
*Mar  8 09:20:44.319:itu2LSC_Rcvd_SIE  chnl=0  MTP2LSC_INITIAL_ALIGNMENT
*Mar  8 09:20:44.319:itu2LSC_Rcvd_SIE  chnl=0  MTP2LSC_INITIAL_ALIGNMENT
*Mar  8 09:20:48.397:itu2LSC_AlignmentComplete  chnl=0
MTP2LSC_INITIAL_ALIGNMENT
```

The following is an example of **debug ss7 mtp2 msu** command output for channel 2. The output for this command can slow traffic under busy conditions, so enter it when there is low traffic. See the MTP 2 specification tables for details about the command output:

```
Router# debug ss7 mtp2 msu 2

*Mar  1 01:01:12.447: MTP2: send MSU Ind  ch=2  len=25
*Mar  1 01:01:12.455: MTP2: rcvd MSU Req  ch=2  len=252
```

Warning **Use this command only for testing problems in a controlled environment. This command can generate significant amounts of output. If there is any significant amount of traffic flow when you issue the command, the processor may slow down so much that RUDP connections fail. This command is recommended for field support personnel only, and is not recommended for use without prior recommendation from Cisco.**

The following is an example of **debug ss7 mtp2 packet** command output for channel 0:

```
Router# debug ss7 mtp2 packet 0

*Mar  1 00:53:00.052: MTP2 incoming trace enabled on channel 0.
*Mar  1 00:53:00.052: MTP2 outgoing trace enabled on channel 0.
*Mar  1 00:53:07.220: ---- Incoming Rudp msg (20 bytes) ----
SM_msg_type    0x00008000
protocol_type  0x0001
msg_ID         0x0001
msg_type       0x0044
channel_ID     0x0000
bearer_ID      0x0000
length         0x0004
data           0x00000001

*Mar  1 00:53:07.224: ---- Outgoing Rudp msg (132 bytes) ----
SM_msg_type    0x00008000
protocol_type  0x0001
msg_ID         0x0001
msg_type       0x0045
channel_ID     0x0000
```

```
bearer_ID      0x0000
length         0x0074
data           0x0000001E 0x00000000 0x00000000 0x00000000
               0x00000000 0x00000000 0x00000000 0x00000000
               0x00000000 0x00000000 0x00000000 0x00000000
               0x00000002 0x00000000 0x00008317 0x00000000
               0x00000002 0x00000000 0x00000008 0x009B5C97
               0x00000000 0x0032A2A7 0x0000061C 0x000000BF
               0x00000000 0x00000000 0x00000006 0x00000000
               0x000000ED

*Mar  1 00:53:11.343: ---- Outgoing Rudp msg (41 bytes) ----
SM_msg_type    0x00008000
protocol_type  0x0001
msg_ID         0x0000
msg_type       0x0011
channel_ID     0x0000
bearer_ID      0x0000
length         0x0019
data           0x8201190A 0x03190A00 0x11F01122 0x33445566
               0x778899AA 0xBBCCDDEE

*Mar  1 00:53:11.351: ---- Incoming Rudp msg (41 bytes) ----
SM_msg_type    0x00008000
protocol_type  0x0001
msg_ID         0x0001
msg_type       0x0010
channel_ID     0x0000
bearer_ID      0x0000
length         0x0019
data           0xB203190A 0x01190A00 0x21F01122 0x33445566
               0x778899AA 0xBBCCDDEE
*Mar  1 00:53:13.739: ---- Incoming Rudp msg (27 bytes) ----
SM_msg_type    0x00008000
protocol_type  0x0001
msg_ID         0x0001
msg_type       0x0010
channel_ID     0x0000
bearer_ID      0x0000
length         0x000B
data           0x9503190A 0x01190A00
```

The following is an example of **debug ss7 mtp2 rcv** command output. See the MTP 2 specification tables for details:

```
Router# debug ss7 mtp2 rcv 0

*Mar  8 09:22:35.160:itu2RC_Stop  chnl=0  MTP2RC_INSERVICE
*Mar  8 09:22:35.164:itu2RC_Start  chnl=0  MTP2RC_IDLE
*Mar  8 09:22:52.565:BSNR not in window
        bsnr=2  bibr=0x80    fsnr=66  fibr=0x80  fsnf=0  fsnl=127  fsnx=0
        fsnt=127

*Mar  8 09:22:52.569:BSNR not in window
        bsnr=2  bibr=0x80    fsnr=66  fibr=0x80  fsnf=0  fsnl=127  fsnx=0
        fsnt=127

*Mar  8 09:22:52.569:AbnormalBSN_flag == TRUE
*Mar  8 09:22:52.569:itu2RC_Stop  chnl=0  MTP2RC_INSERVICE
*Mar  8 09:22:57.561:itu2RC_Start  chnl=0  MTP2RC_IDLE
```

The following is an example of **debug ss7 mtp2 suerm** command output. See the MTP 2 specification tables for details:

```
Router# debug ss7 mtp2 suerm 0

*Mar  8 09:33:51.108:itu2SUERM_Stop  chnl=0  MTP2SUERM_MONITORING
*Mar  8 09:34:00.155:itu2SUERM_Start  chnl=0  MTP2SUERM_IDLE
```

Warning **Use this command only for testing problems in a controlled environment. This command can generate significant amounts of output. If there is any significant amount of traffic flow when you issue the command, the processor may slow down so much that RUDP connections fail. This command is recommended for field support personnel only, and is not recommended for use without prior recommendation from Cisco.**

The following is an example of **debug ss7 mtp2 timer** command output for channel 0:

```
Router# debug ss7 mtp2 timer 0

*Mar  1 01:08:13.738: Timer T7 (ex delay) Start    chnl=0
*Mar  1 01:08:13.762: Timer T7 (ex delay) Stop     chnl=0
*Mar  1 01:08:13.786: Timer T7 (ex delay) Start    chnl=0
*Mar  1 01:08:13.810: Timer T7 (ex delay) Stop     chnl=0
*Mar  1 01:08:43.819: Timer T7 (ex delay) Start    chnl=0
*Mar  1 01:08:43.843: Timer T7 (ex delay) Stop     chnl=0
*Mar  1 01:08:48.603: Timer T7 (ex delay) Start    chnl=0
*Mar  1 01:08:48.627: Timer T7 (ex delay) Stop     chnl=0
*Mar  1 01:09:13.784: Timer T7 (ex delay) Start    chnl=0
*Mar  1 01:09:13.808: Timer T7 (ex delay) Stop     chnl=0
*Mar  1 01:09:13.885: Timer T7 (ex delay) Start    chnl=0
*Mar  1 01:09:13.909: Timer T7 (ex delay) Stop     chnl=0
```

Warning **Use this command only for testing problems in a controlled environment. This command can generate significant amounts of output. If there is any significant amount of traffic flow when you issue the command, the processor may slow down so much that RUDP connections fail. This command is recommended for field support personnel only, and is not recommended for use without prior recommendation from Cisco.**

The following is an example of **debug ss7 mtp2 txc** command output for channel 2. The transmission control is functioning and updating Backward Sequence Numbers (BSNs). See the MTP 2 specification for details:

```
Router# debug ss7 mtp2 txc 2

*Mar  1 01:10:13.831: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:13.831: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:13.831: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:13.839: itu2TXC_PDU2xmit  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:13.863: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:13.863: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:23.603: itu2TXC_PDU2xmit  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:23.627: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:23.627: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:23.631: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:23.631: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:23.635: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:43.900: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:43.900: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:43.900: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
```

```
*Mar  1 01:10:43.908: itu2TXC_PDU2xmit  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:43.928: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVICE
*Mar  1 01:10:43.932: itu2TXC_bsn_update  chnl=2  MTP2TXC_INSERVIC
```

The following MTP2 specification tables explain codes that appear in the command output.

Backhaul Debug Event Codes	Description
0x0	Local processor outage
0x1	Local processor outage recovered
0x2	Entered a congested state
0x3	Exited a congested state
0x4	Physical layer up
0x5	Physical layer down
0x7	Protocol error (see cause code)
0x8	Link alignment lost
0x9	Retransmit buffer full
0xa	Retransmit buffer no longer full
0xc	Remote entered congestion
0xd	Remote exited congestion
0xe	Remote entered processor outage
0xf	Remote exited processor outage

Backhaul Debug Cause Codes	Description
0x0	Cause unknown - default
0x1	Management initiated
0x2	Abnormal BSN (Backward Sequence Number)
0x3	Abnormal FIB (Forward Indicator Bit)
0x4	Congestion discard

Backhaul Debug Reason Codes	Description
0x0	Layer management request
0x1	SUERM (Signal Unit Error Monitor) failure
0x2	Excessively long alignment period
0x3	T7 timer expired
0x4	Physical interface failure
0x5	Two or three invalid BSNs
0x6	Two or three invalid FIBs
0x7	LSSU (Link Status Signal Unit) condition

Backhaul Debug Reason Codes	Description
0x13	SIOs (Service Information Octets) received in Link State Control (LSC)
0x14	Timer T2 expired waiting for SIO
0x15	Timer T3 expired waiting for SIE/SIN
0x16	SIO received in initial alignment control (IAC)
0x17	Proving period failure
0x18	Timer T1 expired waiting for FISU (Fill-In Signal Unit)
0x19	SIN received in the in-service state
0x20	CTS lost
0x25	No resources

Related Commands

Command	Description
debug sse	Displays debug messages for the a Signaling System 7 (SS7) Session Manager.

debug ss7 sm

To display debug messages for the a Signaling System 7 (SS7) Session Manager, use the **debug ss7 sm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ss7 sm [**session** *0-3* | **set** | **timer**]

no debug ss7 sm session

Syntax Description

0-3	Specifies a session ID number 0 to 3.
session	Sets Session Manager session debug.
set	Sets Session Manager debug.
timer	Sets Session Manager timer debug.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)XR and 12.1(1)T	This command was introduced.
12.1(1)T	This command was integrated into Cisco IOS Release 12.1(1)T.
12.2(11)T	This command replaces the **debug ss7 sm session** command. This command was modified with the **session**, **set**, and **timer** keywords. This command was also modified to support up to four Session Manager sessions.

Usage Guidelines

Use this command to watch the Session Manager and Reliable User Data Protocol (RUDP) sessions. The Session Manager is responsible for establishing the RUDP connectivity to the Virtual Switch Controller (VSC).

Support for up to four Session Manager sessions was added. Session Manager sessions are now numbered 0 through 3. This feature changes the CLI syntax, and adds sessions 2 and 3.

Examples

The following is an example of **debug ss7 sm** command output using the **session** keyword. The Session Manager has established the connection (RUDP_CONN_OPEN_SIG) for session 3.

```
Router# debug ss7 sm session 3

*Mar  8 09:37:52.119:SM:rudp signal RUDP_SOFT_RESET_SIG, session = 3
*Mar  8 09:37:58.129:SM:rudp signal RUDP_CONN_RESET_SIG, session = 3
*Mar  8 09:37:58.129:SM:Opening session[0] to 10.5.0.4:8060
*Mar  8 09:37:58.137:SM:rudp signal RUDP_CONN_OPEN_SIG, session = 3
```

The following is an example of **debug ss7 sm session** command output for session 0. The Session Manager has established the connection (RUDP_CONN_OPEN_SIG):

```
Router# debug ss7 sm session 0

*Mar  8 09:37:52.119:SM:rudp signal RUDP_SOFT_RESET_SIG, session = 0
*Mar  8 09:37:58.129:SM:rudp signal RUDP_CONN_RESET_SIG, session = 0
*Mar  8 09:37:58.129:SM:Opening session[0] to 10.5.0.4:8060
*Mar  8 09:37:58.137:SM:rudp signal RUDP_CONN_OPEN_SIG, session = 0
```

Related Commands

Command	Description
encapsulation ss7	Assigns a channel group and selects the DS0 time slots desired for SS7 links.

debug sse

To display information for the silicon switching engine (SSE) processor, use the **debug sse** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sse

no debug sse

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debug sse** command to display statistics and counters maintained by the SSE.

Examples

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

```
Router# debug sse

SSE: IP number of cache entries changed 273 274
SSE: bridging enabled
SSE: interface Ethernet0/0 icb 0x30 addr 0x29 status 0x21A040 protos 0x11
SSE: interface Ethernet0/1 icb 0x33 addr 0x29 status 0x21A040 protos 0x11
SSE: interface Ethernet0/2 icb 0x36 addr 0x29 status 0x21A040 protos 0x10
SSE: interface Ethernet0/3 icb 0x39 addr 0x29 status 0x21A040 protos 0x11
SSE: interface Ethernet0/4 icb 0x3C addr 0x29 status 0x21A040 protos 0x10
SSE: interface Ethernet0/5 icb 0x3F addr 0x29 status 0x21A040 protos 0x11
SSE: interface Hssi1/0 icb 0x48 addr 0x122 status 0x421E080 protos 0x11
SSE: cache update took 316ms, elapsed 320ms
```

The following line indicates that the SSE cache is being updated due to a change in the IP fast-switching cache:

```
SSE: IP number of cache entries changed 273 274
```

The following line indicates that bridging functions were enabled on the SSE:

```
SSE: bridging enabled
```

The following lines indicate that the SSE is now loaded with information about the interfaces:

```
SSE: interface Ethernet0/0 icb 0x30 addr 0x29 status 0x21A040 protos 0x11
SSE: interface Ethernet0/1 icb 0x33 addr 0x29 status 0x21A040 protos 0x11
SSE: interface Ethernet0/2 icb 0x36 addr 0x29 status 0x21A040 protos 0x10
SSE: interface Ethernet0/3 icb 0x39 addr 0x29 status 0x21A040 protos 0x11
SSE: interface Ethernet0/4 icb 0x3C addr 0x29 status 0x21A040 protos 0x10
SSE: interface Ethernet0/5 icb 0x3F addr 0x29 status 0x21A040 protos 0x11
SSE: interface Hssi1/0 icb 0x48 addr 0x122 status 0x421E080 protos 0x11
```

The following line indicates that the SSE took 316 ms of processor time to update the SSE cache. The value of 320 ms represents the total time elapsed while the cache updates were performed.

```
SSE: cache update took 316ms, elapsed 320ms
```

debug ssg ctrl-errors

To display all error messages for control modules, use the **debug ssg ctrl-errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg ctrl-errors

no debug ssg ctrl-errors

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)DC	This command was introduced on the Cisco 6400 node route processor.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

Use this command to show error messages for the control modules. These modules include all those that manage the user authentication and service login and logout (RADIUS, PPP, Subblock, and Accounting). An error message is the result of an error detected during normal execution.

Examples

The following output is generated by using the **debug ssg ctrl-errors** command when a host logs in to and logs out of a service:

```
Router# debug ssg ctrl-errors

Mar 29 13:51:30 [192.168.5.1.15.21] 59:00:15:38:%VPDN-6-AUTHORERR:L2F NAS
LowSlot6 cannot locate a AAA server for Vi6 user User1
Mar 29 13:51:31 [192.168.5.1.15.21] 60:00:15:39:%LINEPROTO-5-UPDOWN:Line
protocol on Interface Virtual-Access6, changed state to down
```

Related Commands

Command	Description
debug ssg ctrl-events	Displays all event messages for control modules.
debug ssg ctrl-packets	Displays packet contents handled by control modules.

debug ssg ctrl-events

To display all event messages for control modules, use the **debug ssg ctrl-events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg ctrl-events

no debug ssg ctrl-events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)DC	This command was introduced on the Cisco 6400 node route processor.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

This command displays event messages for the control modules, which include all modules that manage the user authentication and service login and logout (RADIUS, PPP, Subblock, and Accounting). An event message is an informational message generated during normal execution.

Examples

The following output is generated by the **debug ssg ctrl-events** command when a host logs in to a service:

```
Router# debug ssg ctrl-events

Mar 16 16:20:30 [192.168.6.1.7.141] 799:02:26:51:SSG-CTL-EVN:Service logon is accepted.
Mar 16 16:20:30 [192.168.6.1.7.141] 800:02:26:51:SSG-CTL-EVN:Send cmd 11 to host
172.16.6.13. dst=192.168.100.24:36613
```

Related Commands

Command	Description
debug ssg ctrl-packets	Displays packet contents handled by control modules.
ssg local-forwarding	Displays all error messages for control modules.

debug ssg ctrl-packets

To display packet contents handled by control modules, use the **debug ssg ctrl-packets** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg ctrl-packets

no debug ssg ctrl-packets

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)DC	This command was introduced on the Cisco 6400 node route processor.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

Use this command to show packet messages for the control modules. These modules include all those that manage the user authentication and service login and logout (RADIUS, PPP, Subblock, and Accounting). A packet message displays the contents of a package.

Examples

The following output is generated by using the **debug ssg ctl-packets** command when a host logs out of a service:

```
Router# debug ssg ctrl-packets

Mar 16 16:23:38 [192.168.6.1.7.141] 968:02:30:00:SSG-CTL-PAK:Received Packet:
Mar 16 16:23:38 [192.168.6.1.7.141] 980:02:30:00:SSG-CTL-PAK:Sent packet:
Mar 16 16:23:39 [192.168.6.1.7.141] 991:02:30:00:SSG-CTL-PAK:
Mar 16 16:23:39 [192.168.6.1.7.141] 992:Received Packet:
```

Related Commands

Command	Description
debug ssg ctrl-events	Displays all event messages for control modules.
ssg local-forwarding	Displays all error messages for control modules.

debug ssg data

To display all data-path packets, use the **debug ssg data** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg data

no debug ssg data

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)DC	This command was introduced on the Cisco 6400 node route processor.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

The **debug ssg data** command shows packets for the data modules. These modules include all those that forward data packets (Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), tunneling, fast switching, IP stream, and multicast).

Examples

The following output is generated by using the **debug ssg data** command when a host logs in to and out of a service:

```
Router# debug ssg data

Mar 29 13:45:16 [192.168.5.1.15.21] 45:00:09:24:
SSG-DATA:PS-UP-SetPakOutput=1(Vi6:172.16.5.50->199.199.199.199)
Mar 29 13:45:16 [192.168.5.1.15.21] 46:00:09:24:
SSG-DATA:PS-DN-SetPakOutput=1(Fa0/0/0:171.69.2.132->172.16.5.50)
Mar 29 13:45:16 [192.168.5.1.15.21] 47:00:09:24:
SSG-DATA:FS-UP-SetPakOutput=1(Vi6:172.16.5.50->171.69.43.34)
Mar 29 13:45:16 [192.168.5.1.15.21] 48:00:09:24:
```

Related Commands

Command	Description
debug ssg data-nat	Displays all data-path packets for NAT processing.

debug ssg data-nat

To display all data-path packets for Network Address Translation (NAT) processing, use the **debug ssg data-nat** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg data-nat

no debug ssg data-nat

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)DC	This command was introduced on the Cisco 6400 node route processor.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

The **debug ssg data-nat** command displays packets for the data modules. These modules include all those that forward NAT data packets.

Examples

The following output is generated by using the **debug ssg data-nat** command when a host logs in to and out of a service:

```
Router# debug ssg data-nat

Mar 29 13:43:14 [192.168.5.1.15.21] 35:00:07:21:SSG-DATA:TranslateIP Dst
199.199.199.199->171.69.2.132
Mar 29 13:43:14 [192.168.5.1.15.21] 36:00:07:21:SSG-DATA:TranslateIP Src
171.69.2.132->199.199.199.199
Mar 29 13:43:30 [192.168.5.1.15.21] 39:00:07:38:SSG-DATA:TranslateIP Dst
199.199.199.199->171.69.2.132
Mar 29 13:43:30 [192.168.5.1.15.21] 40:00:07:38:SSG-DATA:TranslateIP Src
171.69.2.132->199.199.199.199
```

Related Commands

Command	Description
debug ssg data	Displays all data-path packets.

debug ssg errors

To display all error messages for the system modules, use the **debug ssg errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg errors

no debug ssg errors

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)DC	This command was introduced on the Cisco 6400 node route processor.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

The **debug ssg errors** command displays error messages for the system modules, which include the basic Cisco IOS and other support modules (such as Object Model, Timeout, and Initialization). An error message is the result of an error detected during normal execution.

Examples

The following output is generated by using the **debug ssg errors** command when a PPP over Ethernet (PPPoE) client logs in with an incorrect password:

```
Router# debug ssg errors

Mar 16 08:46:20 [192.168.6.1.7.141] 225:00:16:06:SSG:SSGDoAccounting:
reg_invoke_do_acct returns FALSE
```

Related Commands

Command	Description
debug ssg events	Displays event messages for system modules.
debug ssg packets	Displays packet contents handled by system modules.

debug ssg events

To display event messages for system modules, use the **debug ssg events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg events

no debug ssg events

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)DC	This command was introduced on the Cisco 6400 node route processor.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

The **debug ssg events** command displays event messages for the system modules, which include the basic Cisco IOS modules and other support modules (such as Object Model, Timeout, and Initialization). An event message is an informational message that appears during normal execution.

Examples

The following output is generated by using the **debug ssg events** command when a PPP over Ethernet (PPPoE) client logs in with the username "username" and the password "cisco":

```
Router# debug ssg events

Mar 16 08:39:39 [192.168.6.1.7.141] 167:00:09:24:%LINK-3-UPDOWN:
Interface Virtual-Access3, changed state to up
Mar 16 08:39:39 [192.168.6.1.7.141] 168:00:09:25:%LINEPROTO-5-UPDOWN:
Line protocol on Interface Virtual-Access3, changed state to up
Mar 16 08:39:40 [192.168.6.1.7.141] 169:00:09:26:%VPDN-6-AUTHORERR:L2F
NAS LowSlot7 cannot locate a AAA server for Vi3 user username
Mar 16 08:39:40 [192.168.6.1.7.141] 170:HostObject::HostObject:size = 256
Mar 16 08:39:40 [192.168.6.1.7.141] 171:HostObject::Reset
Mar 16 08:39:40 [192.168.6.1.7.141] 172:Service List:
Mar 16 08:39:40 [192.168.6.1.7.141] 175:Service = isp-1
```

Related Commands

Command	Description
debug ssg errors	Displays all error messages for the system modules.
debug ssg packets	Displays packet contents handled by system modules.

debug ssg packets

To display packet contents handled by system modules, use the **debug ssg packets** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg packets

no debug ssg packets

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)DC	This command was introduced on the Cisco 6400 node route processor.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

The **debug ssg packets** command displays packet messages for the system modules, which include the basic Cisco IOS and other support modules (such as Object Model, Timeout, Initialization). A packet message displays the contents of a package.

Examples

The following output is generated by using the **debug ssg packets** command when a user is running a Telnet session to 192.168.250.12 and pinging 192.168.250.11:

```
Router# debug ssg packets

19:46:03:SSG-DATA:PS-UP-SetPakOutput=1(Vi2:172.16.17.71->192.168.250.12)
19:46:03:SSG-DATA:PS-UP-SetPakOutput=1(Vi2:172.16.17.71->192.168.250.12)
19:46:03:SSG-DATA:PS-UP-SetPakOutput=1(Vi3:172.16.17.72->192.168.250.12)
19:46:03:SSG-DATA:PS-UP-SetPakOutput=1(Vi2:172.16.17.71->192.168.250.12)
19:46:03:SSG-DATA:PS-UP-SetPakOutput=1(Vi2:172.16.17.71->192.168.250.12)
19:46:03:SSG-DATA:PS-UP-SetPakOutput=1(Vi2:172.16.17.71->192.168.250.12)
19:46:03:SSG-DATA:PS-UP-SetPakOutput=1(Vi3:172.16.17.72->192.168.250.11)
```

Related Commands

Command	Description
debug ssg errors	Displays all error messages for the system modules.
debug ssg events	Displays event messages for system modules.

debug ssg port-map

To display debug messages for port-mapping, use the **debug ssg port-map** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg port-map {**events** | **packets**}

no debug ssg port-map {**events** | **packets**}

Syntax Description

events	Displays messages for port-map events: create and remove.
packets	Displays port-map packet contents and port address translations.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)B	This command was introduced on the Cisco 6400 series.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(2)XB	This command was integrated into Cisco IOS Release 12.2(2)XB.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.

Usage Guidelines

This command displays debug messages for the creation of port maps.

Examples

Using the **debug ssg port-map** command generates the following output when a subscriber logs in to a service:

```
Router# debug ssg port-map event
SSG port-map events debugging is on

Router# show debug

SSG:
  SSG port-map events debugging is on
Router#
00:46:09:SSG-PMAP:Changing state of port-bundle 70.13.60.3:65 from FREE to RESERVED
00:46:09:SSG-PMAP:Changing state of port-bundle 70.13.60.3:65 from RESERVED to INUSE
00:46:10:%LINEPROTO-5-UPDOWN:Line protocol on Interface Virtual-Access2, changed state to
up
Router#
00:46:25:SSG-PMAP:Allocating new port-mapping:[4148<->1040] for port-bundle 70.13.60.3:65
00:46:29:SSG-PMAP:Allocating new port-mapping:[4149<->1041] for port-bundle 70.13.60.3:65
00:46:31:SSG-PMAP:Allocating new port-mapping:[4150<->1042] for port-bundle 70.13.60.3:65
00:46:31:SSG-PMAP:Allocating new port-mapping:[4151<->1043] for port-bundle 70.13.60.3:65
00:46:31:SSG-PMAP:Allocating new port-mapping:[4152<->1044] for port-bundle 70.13.60.3:65
```

```
Router# debug ssg port-map packets

SSG port-map packets debugging is on
Router#
00:51:55:SSG-PMAP:forwarding non-TCP packet
00:51:55:SSG-PMAP:forwarding packet
00:51:55:SSG-PMAP:forwarding non-TCP packet
00:51:55:SSG-PMAP:forwarding packet
00:51:55:SSG-PMAP:forwarding non-TCP packet
00:52:06:SSG-PMAP:srcip:70.13.6.100 srcport:8080  dstip:70.13.60.3 dstport:1044
00:52:06:SSG-PMAP:TCP flags:5011  Seq no:1162897784 Ack no:-1232234715
00:52:06:SSG-PMAP:received TCP-FIN packet
00:52:10:SSG-PMAP:cef:packet bound for default n/w
00:52:10:SSG-PMAP:Checking port-map ACLs
00:52:10:SSG-PMAP:Port-map ACL check passed
00:52:10:SSG-PMAP:cef:punting TCP-SYN packet to process
00:52:10:SSG-PMAP:packet bound for default n/w
00:52:10:SSG-PMAP:fast:punting TCP-SYN packet to process
00:52:10:SSG-PMAP:packet bound for default n/w
00:52:10:SSG-PMAP:translating source address from 10.3.6.1 to 70.13.60.3
00:52:10:SSG-PMAP:translating source port from 4158 to 1040
00:52:10:SSG-PMAP:srcip:70.13.6.100 srcport:8080  dstip:70.13.60.3 dstport:1040
00:52:10:SSG-PMAP:TCP flags:6012  Seq no:1186352744 Ack no:-1232047701
00:52:10:SSG-PMAP:translating destination address from 70.13.60.3 to 10.3.6.1
00:52:10:SSG-PMAP:translating destination port from 1040 to 4158
```

Related Commands

Command	Description
show ssg port-map ip	Displays information on a particular port bundle.
show ssg port-map status	Displays information on port bundles.

debug ssg tcp-redirect

To turn on debug information for the Service Selection Gateway (SSG) Transport Control Protocol (TCP) Redirect for Services feature, use the **debug ssg tcp-redirect** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ssg tcp-redirect {**packet** | **error** | **event**}

no debug ssg tcp-redirect {**packet** | **error** | **event**}

Syntax Description

packet	Displays redirection information and any changes made to a packet when it is due for redirection.
error	Displays any SSG TCP redirect errors.
event	Displays any major SSG TCP redirect events or state changes.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(4)B	This command was introduced in Cisco IOS Release 12.2(4)B.
12.2(2)XB	This command was integrated in Cisco IOS Release 12.2(2)XB.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.

Usage Guidelines

Use this command to turn on debug information for the SSG TCP Redirect for Services feature. Use the **packet** keyword to display redirection information and any changes made to a packet when it is due for redirection. Use the **error** keyword to display any SSG TCP redirect errors. Use the **event** keyword to display any major SSG TCP redirect events or state changes.

Note This command replaces the **debug ssg packets** command.

Examples

The following example shows how to display redirection information and any changes made to a packet when it is due for redirection:

```
Router# debug ssg tcp-redirect packet
```

Direction of the packet "-Up" indicates upstream packets from an SSG user, while "-Down" indicates downstream packets sent to a user:

```
07:13:15:SSG-REDIR-PKT:-Up:unauthorised user at 111.0.0.2 redirected to 9.2.36.253,8080

07:13:15:SSG-REDIR-PKT:-Down:TCP-RST Rxd for user at 111.0.0.2, port 11114

07:13:15:SSG-REDIR-PKT:-Down:return remap for user at 111.0.0.2 redirected from 9.2.36.25
```

The following example shows how to display any SSG TCP redirect errors:

```
Router# debug ssg tcp-redirect error

07:15:20:SSG-REDIR-ERR:-Up:Packet from 172.0.0.2:11114 has different destination from
stored connection
```

The following example shows how to display any major SSG TCP redirect events or state changes:

```
Router# debug ssg tcp-redirect event
```

Upstream packets from users are redirected:

```
06:45:51:SSG-TCP-REDIR:-Up:created new remap entry for unauthorised user at 172.16.0.2
06:45:51:                  Redirect server set to  10.2.36.253,8080
06:45:51:                  Initial src/dest port mapping 11094<->23
06:45:51:SSG-REDIR-EVT: Freeing tcp-remap connections
06:46:21:SSG-REDIR-EVT:Host at 111.0.0.2, connection port  11094  timed out
06:46:21:SSG-REDIR-EVT: Unauthenticated user remapping for 172.16.0.2 removed
```

A host is being activated:

```
06:54:09:SSG-REDIR-EVT:- New Host at 172.16.0.2 set for default initial captivation
06:54:09:SSG-REDIR-EVT:- New Host at 172.16.0.2 set for default advertising captivation
```

Initial captivation begins:

```
06:59:32:SSG-REDIR-EVT:-Up:initial captivate got packet at start of connection (from
111.0.0.2)
06:59:32:SSG-REDIR-EVT:-Up:user at 111.0.0.2 starting initial captivation
06:59:32:SSG-REDIR-EVT:- Up:created new redirect connection and server for user at
111.0.0.2
06:59:32:         Redirect server set to  10.64.131.20,8000
06:59:32:         Initial src/dest port mapping 11109<->80
06:59:48:SSG-REDIR-EVT:-Up:initial captivate got packet at start of connection (from
111.0.0.2)
06:59:48:SSG-REDIR-EVT:-Up:initial captivate timed out for user at 172.16.0.2
06:59:48:SSG-REDIR-EVT:Removing server 10.64.131.20:8000 for host 172.16.0.2
```

Advertising captivation begins:

```
06:59:48:SSG-REDIR-EVT:Removing redirect map for host 172.16.0.2
06:59:48:SSG-REDIR-EVT:-Up:advert captivate got packet at start of connection (from
111.0.0.2)
06:59:48:SSG-REDIR-EVT:-Up:user at 111.0.0.2 starting advertisement captivation
06:59:48:SSG-REDIR-EVT:- Up:created new redirect connection and server for user at
111.0.0.2
06:59:48:         Redirect server set to  10.64.131.20,8000
06:59:48:         Initial src/dest port mapping 11110<->80
```

Related Commands

Command	Description
show ssg tcp-redirect group	Displays information about the captive portal groups and the networks associated with the captive portal groups.
show tcp-redirect mappings	Displays information about the TCP redirect mappings for hosts within your system.
ssg enable	Enables SSG.
ssg tcp-redirect	Enables SSG TCP redirect and enters SSG-redirect mode.

debug sss aaa authorization event

To display messages about authentication, authorization, and accounting (AAA) authorization events that are part of normal call establishment, use the **debug sss aaa authorization event** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sss aaa authorization event

no debug sss aaa authorization event

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(13)T	This command was introduced.

Examples

The following is sample output of several Subscriber Service Switch **debug** commands including the **debug sss aaa authorization event** command. The reports from these commands should be sent to technical personnel at Cisco Systems for evaluation.

```
Router# debug sss event
Router# debug sss error
Router# debug sss state
Router# debug sss aaa authorization event
Router# debug sss aaa authorization fsm

SSS:
  SSS events debugging is on
  SSS error debugging is on
  SSS fsm debugging is on
  SSS AAA authorization event debugging is on
  SSS AAA authorization FSM debugging is on

*Mar  4 21:33:18.248: SSS INFO: Element type is Access-Type, long value is 3
*Mar  4 21:33:18.248: SSS INFO: Element type is Switch-Id, long value is -1509949436
*Mar  4 21:33:18.248: SSS INFO: Element type is Nasport, ptr value is 6396882C
*Mar  4 21:33:18.248: SSS INFO: Element type is AAA-Id, long value is 7
*Mar  4 21:33:18.248: SSS INFO: Element type is AAA-ACCT_ENBL, long value is 1
*Mar  4 21:33:18.248: SSS INFO: Element type is AccIe-Hdl, ptr value is 78000006
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Event service-request, state changed from
wait-for-req to wait-for-auth
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Handling Policy Authorize (1 pending sessions)
*Mar  4 21:33:18.248: SSS PM [uid:7]: Need the following key: Unauth-User
*Mar  4 21:33:18.248: SSS PM [uid:7]: Received Service Request
```

```
*Mar  4 21:33:18.248: SSS PM [uid:7]: Event <need keys>, State: initial-req to
need-init-keys
*Mar  4 21:33:18.248: SSS PM [uid:7]: Policy reply - Need more keys
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Got reply Need-More-Keys from PM
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Event policy-or-mgr-more-keys, state changed from
wait-for-auth to wait-for-req
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Handling More-Keys event
*Mar  4 21:33:20.256: SSS INFO: Element type is Unauth-User, string value is
nobody2@xyz.com
*Mar  4 21:33:20.256: SSS INFO: Element type is AccIe-Hdl, ptr value is 78000006
*Mar  4 21:33:20.256: SSS INFO: Element type is AAA-Id, long value is 7
*Mar  4 21:33:20.256: SSS INFO: Element type is Access-Type, long value is 0
*Mar  4 21:33:20.256: SSS MGR [uid:7]: Event service-request, state changed from
wait-for-req to wait-for-auth
*Mar  4 21:33:20.256: SSS MGR [uid:7]: Handling Policy Authorize (1 pending sessions)
*Mar  4 21:33:20.256: SSS PM [uid:7]: Received More Initial Keys
*Mar  4 21:33:20.256: SSS PM [uid:7]: Event <rcvd keys>, State: need-init-keys to
check-auth-needed
*Mar  4 21:33:20.256: SSS PM [uid:7]: Handling Authorization Check
*Mar  4 21:33:20.256: SSS PM [uid:7]: Event <send auth>, State: check-auth-needed to
authorizing
*Mar  4 21:33:20.256: SSS PM [uid:7]: Handling AAA service Authorization
*Mar  4 21:33:20.256: SSS PM [uid:7]: Sending authorization request for 'xyz.com'
*Mar  4 21:33:20.256: SSS AAA AUTHOR [uid:7]:Event <make request>, state changed from idle
to authorizing
*Mar  4 21:33:20.256: SSS AAA AUTHOR [uid:7]:Authorizing key xyz.com
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:AAA request sent for key xyz.com
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Received an AAA pass
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Event <found service>, state changed from
authorizing to complete
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Found service info for key xyz.com
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Event <free request>, state changed from
complete to terminal
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Free request
*Mar  4 21:33:20.264: SSS PM [uid:7]: Event <found>, State: authorizing to end
*Mar  4 21:33:20.264: SSS PM [uid:7]: Handling Service Direction
*Mar  4 21:33:20.264: SSS PM [uid:7]: Policy reply - Forwarding
*Mar  4 21:33:20.264: SSS MGR [uid:7]: Got reply Forwarding from PM
*Mar  4 21:33:20.264: SSS MGR [uid:7]: Event policy-start-service-fsp, state changed from
wait-for-auth to wait-for-service
*Mar  4 21:33:20.264: SSS MGR [uid:7]: Handling Connect-Forwarding-Service event
*Mar  4 21:33:20.272: SSS MGR [uid:7]: Event service-fsp-connected, state changed from
wait-for-service to connected
*Mar  4 21:33:20.272: SSS MGR [uid:7]: Handling Forwarding-Service-Connected event
```

Related Commands

Command	Description
debug sss aaa authorization fsm	Displays information about AAA authorization state changes.
debug sss error	Displays diagnostic information about errors that may occur during Subscriber Service Switch call setup.
debug sss event	Displays diagnostic information about Subscriber Service Switch call setup events.
debug sss fsm	Displays diagnostic information about the Subscriber Service Switch call setup state.

debug sss aaa authorization fsm

To display information about authentication, authorization, and accounting (AAA) authorization state changes, use the **debug sss aaa authorization fsm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sss aaa authorization fsm

no debug sss aaa authorization fsm

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(13)T	This command was introduced.

Examples

The following example shows how to enter this command. See the "Examples" section of the **debug sss aaa authorization event** command page for an example of output.

```
Router# debug sss aaa authorization fsm
```

Related Commands

Command	Description
debug sss aaa authorization event	Displays messages about AAA authorization events that are part of normal call establishment.
debug sss error	Displays diagnostic information about errors that may occur during Subscriber Service Switch call setup.
debug sss event	Displays diagnostic information about Subscriber Service Switch call setup events.
debug sss fsm	Displays diagnostic information about the Subscriber Service Switch call setup state.

debug sss error

To display diagnostic information about errors that may occur during Subscriber Service Switch call setup, use the **debug sss error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sss error

no debug sss error

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(13)T	This command was introduced.

Examples

The following example shows how to enter this command. See the "Examples" section of the **debug sss aaa authorization event** command page for an example of output.

```
Router# debug sss error
```

Related Commands[

Command	Description
debug sss aaa authorization event	Displays messages about AAA authorization events that are part of normal call establishment.
debug sss aaa authorization fsm	Displays information about AAA authorization state changes.
debug sss event	Displays diagnostic information about Subscriber Service Switch call setup events.
debug sss fsm	Displays diagnostic information about the Subscriber Service Switch call setup state.

debug sss event

To display diagnostic information about Subscriber Service Switch call setup events, use the **debug sss event** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sss event

no debug sss event

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(13)T	This command was introduced.

Examples

The following example shows how to enter this command. See the "Examples" section of the **debug sss aaa authorization event** command page for an example of output.

```
Router# debug sss event
```

Related Commands

Command	Description
debug sss aaa authorization event	Displays messages about AAA authorization events that are part of normal call establishment.
debug sss aaa authorization fsm	Displays information about AAA authorization state changes.
debug sss error	Displays diagnostic information about errors that may occur during Subscriber Service Switch call setup.
debug sss fsm	Displays diagnostic information about the Subscriber Service Switch call setup state.

debug sss fsm

To display diagnostic information about the Subscriber Service Switch call setup state, use the **debug sss fsm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sss fsm

no debug sss fsm

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(13)T	This command was introduced.

Examples

The following example shows how to enter this command. See the "Examples" section of the **debug sss aaa authorization event** command page for an example of output.

```
Router# debug sss fsm
```

debug standby

To display Hot Standby Protocol state changes, use the **debug standby** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug standby

no debug standby

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The **debug standby** command displays Hot Standby Protocol state changes and debugging information regarding transmission and receipt of Hot Standby Protocol packets. Use this command to determine whether hot standby routers recognize one another and take the proper actions.

Examples

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

```
Router# debug standby

SB: Ethernet0 state Virgin -> Listen
SB: Starting up hot standby process
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB: Ethernet0 state Listen -> Speak
SB:Ethernet0 Hello out 192.168.72.20 Speak pri 100 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello out 192.168.72.20 Speak pri 100 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello out 192.168.72.20 Speak pri 100 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB: Ethernet0 state Speak -> Standby
SB:Ethernet0 Hello out 192.168.72.20 Standby pri 100 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello out 192.168.72.20 Standby pri 100 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello out 192.168.72.20 Standby pri 100 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Active pri 90 hel 3 hol 10 ip 192.168.72.29
SB: Ethernet0 Coup out 192.168.72.20 Standby pri 100 hel 3 hol 10 ip 192.168.72.29
SB: Ethernet0 state Standby -> Active
SB:Ethernet0 Hello out 192.168.72.20 Active pri 100 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Speak pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello out 192.168.72.20 Active pri 100 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello in 192.168.72.21 Speak pri 90 hel 3 hol 10 ip 192.168.72.29
SB:Ethernet0 Hello out 192.168.72.20 Active pri 100 hel 3 hol 10 ip 192.168.72.29
```

Table 233 describes the significant fields shown in the display.

Table 233 debug standby Field Descriptions

Field	Description
SB	Abbreviation for “standby.”
Ethernet0	Interface on which a Hot Standby packet was sent or received.
Hello in	Hello packet received from the specified IP address.
Hello out	Hello packet sent from the specified IP address.
pri	Priority advertised in the hello packet.
hel	Hello interval advertised in the hello packet.
hol	Hold-down interval advertised in the hello packet.
ip <*address*>	Hot Standby group IP address advertised in the hello packet.
state	Transition from one state to another.
Coup out <*address*>	Coup packet sent by the router from the specified IP address.

The following line indicates that the router is initiating the Hot Standby Protocol. The **standby ip** interface configuration command enables Hot Standby.

```
SB: Starting up hot standby process
```

The following line indicates that a state transition occurred on the interface:

```
SB: Ethernet0 state Listen -> Speak
```

debug standby events icmp

To display debug messages for the Hot Standby Router Protocol (HSRP) Internet Control Message Protocol (ICMP) redirects filter, use the **debug standby events icmp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug standby events icmp

no debug standby events icmp

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

This command helps you determine whether HSRP is filtering an outgoing ICMP redirect message.

Examples

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

```
Router# debug standby events icmp

10:35:20: SB: changing ICMP redirect sent to 20.0.0.4 for dest 30.0.0.2
10:35:20: SB:   gw 20.0.0.2 -> 20.0.0.12, src 20.0.0.11
10:35:20: SB: Use HSRP virtual address 20.0.0.11 as ICMP src
```

If the router being redirected to is passive (HSRP enabled but no active groups), the following debug message is displayed:

```
10:41:22: SB: ICMP redirect not sent to 20.0.0.4 for dest 40.0.0.3
10:41:22: SB:  20.0.0.3 does not contain an active HSRP group
```

If HSRP could not uniquely determine the gateway used by the host, then the following message is displayed:

```
10:43:08: SB: ICMP redirect not sent to 20.0.0.4 for dest 30.0.0.2
10:43:08: SB: could not uniquely determine IP address for mac 00d0.bbd3.bc22
```

The following messages are also displayed if the **debug ip icmp** command is enabled, in which case the message prefix is changed:

```
10:39:09: ICMP: HSRP changing redirect sent to 20.0.0.4 for dest 30.0.0.2
10:39:09: ICMP:   gw 20.0.0.2 -> 20.0.0.12, src 20.0.0.11
10:39:09: ICMP: Use HSRP virtual address 20.0.0.11 as ICMP src
10:39:09: ICMP: redirect sent to 20.0.0.4 for dest 30.0.0.2, use gw 20.0.0.12
```

Related Commands

Command	Description
debug ip icmp	Displays information on ICMP transactions.

debug stun packet

To display information on packets traveling through the serial tunnel (STUN) links, use the **debug stun packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug stun packet [*group*] [*address*]

no debug stun packet [*group*] [*address*]

Syntax Description

group	(Optional) A decimal integer assigned to a group. Using this option limits output to packets associated with the specified STUN group.
address	(Optional) The output is further limited to only those packets containing the specified STUN address. The *address* argument is in the appropriate format for the STUN protocol running for the specified group.

Command Modes

Privileged EXEC

Usage Guidelines

Because using this command is processor intensive, it is best to use it after regular business hours, rather than in a production environment. It is also best to turn this command on by itself, rather than use it in conjunction with other **debug** commands.

Examples

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

```
router# debug stun packet

STUN sdlc: 0:00:04 Serial3        NDI: (0C2/008) U: SNRM    PF:1
STUN sdlc: 0:00:04 Serial3        NDI: (0C2/008) U: SNRM    PF:1
STUN sdlc: 0:00:01 Serial3        SDI: (0C2/008) U: UA      PF:1
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:000
STUN sdlc: 0:00:00 Serial3        NDI: (0C2/008) I:         PF:1  NR:000  NS:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) I:         PF:1  NR:001  NS:000
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:001
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:001
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:001
STUN sdlc: 0:00:00 Serial3        SDI: (0C2/008) S: RR      PF:1  NR:001
```

X1 type of packet
X2 type of packet
X3 type of packet
S2563

The following line describes an X1 type of packet:

```
STUN sdlc: 0:00:04 Serial3         NDI: (0C2/008) U: SNRM    PF:1
```

Table 234 describes the significant fields in this line of **debug stun packet** output.

Table 234 debug stun packet Field Descriptions

Field	Description
STUN sdlc:	Indication that the STUN feature is providing the information.
0:00:04	Time elapsed since receipt of the previous packet.
Serial3	Interface type and unit number reporting the event.
NDI:	Type of cloud separating the SDLC end nodes. Possible values are as follows: • NDI—Network input • SDI—Serial link
0C2	SDLC address of the SDLC connection.
008	Modulo value of 8.
U: SNRM	Frame type followed by the command or response type. In this case it is an Unnumbered frame that contains a Set Normal Response Mode (SNRM) command. The possible frame types are as follows: • I—Information frame • S—Supervisory frame. The possible commands and responses are: RR (Receive Ready), RNR (Receive Not Ready), and REJ (Reject). • U—Unnumbered frame. The possible commands are: UI (Unnumbered Information), SNRM, DISC/RD (Disconnect/Request Disconnect), SIM/RIM, XID Exchange Identification), TEST. The possible responses are UA (unnumbered acknowledgment), DM (Disconnected Mode), and FRMR (Frame Reject Mode)
PF:1	Poll/Final bit. Possible values are as follows: • 0—Off • 1—On

The following line of output describes an X2 type of packet:

```
STUN sdlc: 0:00:00 Serial3         SDI: (0C2/008) S: RR      PF:1 NR:000
```

All the fields in the previous line of output match those for an X1 type of packet, except the last field, which is additional. NR:000 indicates a receive count of 0; the range for the receive count is 0 to 7.

The following line of output describes an X3 type of packet:

```
STUN sdlc: 0:00:00 Serial3         SDI: (0C2/008) S:I PF:1 NR:000 NS:000
```

All fields in the previous line of output match those for an X2 type of packet, except the last field, which is additional. NS:000 indicates a send count of 0; the range for the send count is 0 to 7.

debug sw56

To display debug information for switched 56K services, use the **debug sw56** command in privileged EXEC mode.

debug sw56

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

debug syscon perfdata

To display messages related to performance data collection, use the **debug syscon perfdata** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug syscon perfdata

no debug syscon perfdata

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command is primarily useful to your technical support representative.

Examples

The following is sample output from the **debug syscon perfdata** command. In this example, the CallFail poll group is configured and applied to shelf 1111. The system determines when the next polling cycle should occur and polls the shelf at the appropriate time. The data is stored in the file CallFail.891645120, and an older file is deleted.

```
Router# debug syscon perfdata

PERF: Applying 'CallFail' to shelf 1111
PERF: Setting up objects for SNMP polling: 'CallFail', shelf 1111
PERF: year hours mins secs msecs = 1998 15 11 1 5
PERF: Start 'CallFail' timer, next cycle in 0 mins, 59 secs
PERF: Timer event:  CallFail, 4 minutes
PERF: Polling 'CallFail', shelf 1111, pc 60AEFDF0
PERF: SNMP resp: Type 6, 'CallFail', shelf 1111, error_st 0
PERF: Logged polled data to disk0:/performance/shelf-1111/CallFail.891645120
PERF: Deleted disk0:/performance/shelf-1111/CallFail.891637469
```

debug syscon sdp

To display messages related to the Shelf Discovery Protocol (SDP), use the **debug syscon sdp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug syscon sdp

no debug syscon sdp

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Use this command to display information about SDP packets exchanged between the shelf and the system controller.

Examples

The following sample output from the **debug syscon sdp** command shows the system controller discovering a managed shelf. In the first few lines, the system controller receives a hello packet from shelf 99 at 172.23.66.106. The system controller responds with a hello packet. When the shelf sends another hello packet, the system controller resets the timer and sends another packet.

```
Syscon# debug syscon sdp

SYSCTLR: Hello packet received via UDP from 172.23.66.106
%SYSCTLR-6-SHELF_ADD: Shelf 99 discovered located at address 172.23.66.106
Hello packet sent to the RS located at 172.23.66.106
SYSCTLR: Hello packet received via UDP from 172.23.66.106
Timer for shelf 99 updated, shelf is alive
Hello packet sent to the RS located at 172.23.66.106
```

The following sample output from the **debug syscon sdp** command shows the shelf contacting the system controller. The shelf sends a hello packet to the system controller at 172.23.66.111. The system controller responds with the autoconfiguration commands. The remaining lines show the Hello packets were exchanged between the shelf and the system controller.

```
Shelf# debug syscon sdp

SYSCTLR: Hello packet sent to the SYSCTLR at 172.23.66.111
SYSCTLR: Command packet received from SYSCTLR
Feb 24 17:24:16.713: %SHELF-6-SYSCTLR_ESTABLISHED: Configured via system controller
located at 172.23.66.111
SYSCTLR: Rcvd HELLO from SYSCTLR at 172.23.66.111
SYSCTLR: Hello packet sent to the SYSCTLR at 172.23.66.111
SYSCTLR: Rcvd HELLO from SYSCTLR at 172.23.66.111
```

debug syslog-server

To display information about the syslog server process, use the **debug syslog-server** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug syslog-server

no debug syslog-server

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command outputs a message every time the syslog server receives a message. It also displays information about subfile creation, removal, and renaming.

Use this command when subfiles are not being created as configured or data is not being written to subfiles. This command is also useful for detecting syslog file size mismatches.

Examples

The following sample display shows output when the following command has been added to the configuration:

```
logging syslog-server 10 3 syslogs
```

This example shows the files being created. Use the **dir disk0:/syslogs.dir** command to display the contents of the newly created directory.

```
Router# debug syslog-server

SYSLOG_SERVER:Syslog file syslogs
SYSLOG_SERVER:Directory disk0:/syslogs.dir created.
SYSLOG_SERVER:Syslog file syslogs created successfully.
```

When a syslog message is received, the router checks to determine if the current file will be too large when the new data is added. In this example, two messages are added to the file.

```
SYSLOG_SERVER: Configured size : 10240 bytes
Current size : 0 bytes
Data size : 68 bytes
New size : 68 bytes
SYSLOG_SERVER: Wrote 68 bytes successfully.
SYSLOG_SERVER: Configured size : 10240 bytes
Current size : 68 bytes
Data size : 61 bytes
New size : 129 bytes
SYSLOG_SERVER: Wrote 61 bytes successfully.
```

Table 235 describes the significant fields shown in the display.

Table 235 debug syslog-server Field Descriptions

Field	Description
Configured size	Maximum subfile size, as set in the **logging syslog-server** command.
Current size	Size of the current subfile before the new message is added.
Data size	Size of the syslog message.
New size	Size of the current subfile after the syslog message is added.

The following output indicates that the current file is too full to fit the next syslog message. The oldest subfile is removed, and the remaining files are renamed. A new file is created and opened for writing syslog messages.

```
SYSLOG_SERVER:Last archive subfile disk0:/syslogs.dir/syslogs.2 removed.
SYSLOG_SERVER: Subfile disk0:/syslogs.dir/syslogs.1 renamed as
disk0:/syslogs.dir/syslogs.2.
SYSLOG_SERVER:subfile disk0:/syslogs.dir/syslogs.cur renamed as
disk0:/syslogs.dir/syslogs.1.
SYSLOG_SERVER:Current subfile disk0:/syslogs.dir/syslogs.cur has been opened.
```

debug tacacs

To display information associated with the TACACS, use the **debug tacacs** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug tacacs

no debug tacacs

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

TACACS is a distributed security system that secures networks against unauthorized access. Cisco supports TACACS under the authentication, authorization, and accounting (AAA) security system.

Use the **debug aaa authentication** command to get a high-level view of login activity. When TACACS is used on the router, you can use the **debug tacacs** command for more detailed debugging information.

Examples

The following is sample output from the **debug aaa authentication** command for a TACACS login attempt that was successful. The information indicates that TACACS+ is the authentication method used.

```
Router# debug aaa authentication

14:01:17: AAA/AUTHEN (567936829): Method=TACACS+
14:01:17: TAC+: send AUTHEN/CONT packet
14:01:17: TAC+ (567936829): received authen response status = PASS
14:01:17: AAA/AUTHEN (567936829): status = PASS
```

The following is sample output from the **debug tacacs** command for a TACACS login attempt that was successful, as indicated by the status PASS:

```
Router# debug tacacs

14:00:09: TAC+: Opening TCP/IP connection to 192.168.60.15 using source 10.116.0.79
14:00:09: TAC+: Sending TCP/IP packet number 383258052-1 to 192.168.60.15 (AUTHEN/START)
14:00:09: TAC+: Receiving TCP/IP packet number 383258052-2 from 192.168.60.15
14:00:09: TAC+ (383258052): received authen response status = GETUSER
14:00:10: TAC+: send AUTHEN/CONT packet
14:00:10: TAC+: Sending TCP/IP packet number 383258052-3 to 192.168.60.15 (AUTHEN/CONT)
14:00:10: TAC+: Receiving TCP/IP packet number 383258052-4 from 192.168.60.15
14:00:10: TAC+ (383258052): received authen response status = GETPASS
14:00:14: TAC+: send AUTHEN/CONT packet
14:00:14: TAC+: Sending TCP/IP packet number 383258052-5 to 192.168.60.15 (AUTHEN/CONT)
14:00:14: TAC+: Receiving TCP/IP packet number 383258052-6 from 192.168.60.15
14:00:14: TAC+ (383258052): received authen response status = PASS
14:00:14: TAC+: Closing TCP/IP connection to 192.168.60.15
```

The following is sample output from the **debug tacacs** command for a TACACS login attempt that was unsuccessful, as indicated by the status FAIL:

```
Router# debug tacacs

13:53:35: TAC+: Opening TCP/IP connection to 192.168.60.15 using source
192.48.0.79
13:53:35: TAC+: Sending TCP/IP packet number 416942312-1 to 192.168.60.15
(AUTHEN/START)
13:53:35: TAC+: Receiving TCP/IP packet number 416942312-2 from 192.168.60.15
13:53:35: TAC+ (416942312): received authen response status = GETUSER
13:53:37: TAC+: send AUTHEN/CONT packet
13:53:37: TAC+: Sending TCP/IP packet number 416942312-3 to 192.168.60.15
(AUTHEN/CONT)
13:53:37: TAC+: Receiving TCP/IP packet number 416942312-4 from 192.168.60.15
13:53:37: TAC+ (416942312): received authen response status = GETPASS
13:53:38: TAC+: send AUTHEN/CONT packet
13:53:38: TAC+: Sending TCP/IP packet number 416942312-5 to 192.168.60.15
(AUTHEN/CONT)
13:53:38: TAC+: Receiving TCP/IP packet number 416942312-6 from 192.168.60.15
13:53:38: TAC+ (416942312): received authen response status = FAIL
13:53:40: TAC+: Closing TCP/IP connection to 192.168.60.15
```

Related Commands

Command	Description
debug aaa accounting	Displays information on accountable events as they occur.
debug aaa authentication	Displays information on AAA/TACACS+ authentication.

debug tacacs events

To display information from the TACACS+ helper process, use the **debug tacacs events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug tacacs events

no debug tacacs events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debug tacacs events** command only in response to a request from service personnel to collect data when a problem has been reported.

Caution Use the **debug tacacs events** command with caution because it can generate a substantial amount of output.

The TACACS protocol is used on routers to assist in managing user accounts. TACACS+ enhances the TACACS functionality by adding security features and cleanly separating out the authentication, authorization, and accounting (AAA) functionality.

Examples

The following is sample output from the **debug tacacs events** command. In this example, the opening and closing of a TCP connection to a TACACS+ server are shown, and the bytes read and written over the connection and the TCP status of the connection:

```
Router# debug tacacs events

%LINK-3-UPDOWN: Interface Async2, changed state to up
00:03:16: TAC+: Opening TCP/IP to 192.168.58.104/1049 timeout=15
00:03:16: TAC+: Opened TCP/IP handle 0x48A87C to 192.168.58.104/1049
00:03:16: TAC+: periodic timer started
00:03:16: TAC+: 192.168.58.104 req=3BD868 id=-1242409656 ver=193 handle=0x48A87C (ESTAB)
expire=14 AUTHEN/START/SENDAUTH/CHAP queued
00:03:17: TAC+: 192.168.58.104 ESTAB 3BD868 wrote 46 of 46 bytes
00:03:22: TAC+: 192.168.58.104 CLOSEWAIT read=12 wanted=12 alloc=12 got=12
00:03:22: TAC+: 192.168.58.104 CLOSEWAIT read=61 wanted=61 alloc=61 got=49
00:03:22: TAC+: 192.168.58.104 received 61 byte reply for 3BD868
00:03:22: TAC+: req=3BD868 id=-1242409656 ver=193 handle=0x48A87C (CLOSEWAIT) expire=9
AUTHEN/START/SENDAUTH/CHAP processed
00:03:22: TAC+: periodic timer stopped (queue empty)
00:03:22: TAC+: Closing TCP/IP 0x48A87C connection to 192.168.58.104/1049
00:03:22: TAC+: Opening TCP/IP to 192.168.58.104/1049 timeout=15
00:03:22: TAC+: Opened TCP/IP handle 0x489F08 to 192.168.58.104/1049
00:03:22: TAC+: periodic timer started
00:03:22: TAC+: 192.168.58.104 req=3BD868 id=299214410 ver=192 handle=0x489F08 (ESTAB)
expire=14 AUTHEN/START/SENDPASS/CHAP queued
```

```
00:03:23: TAC+: 192.168.58.104 ESTAB 3BD868 wrote 41 of 41 bytes
00:03:23: TAC+: 192.168.58.104 CLOSEWAIT read=12 wanted=12 alloc=12 got=12
00:03:23: TAC+: 192.168.58.104 CLOSEWAIT read=21 wanted=21 alloc=21 got=9
00:03:23: TAC+: 192.168.58.104 received 21 byte reply for 3BD868
00:03:23: TAC+: req=3BD868 id=299214410 ver=192 handle=0x489F08 (CLOSEWAIT) expire=13
AUTHEN/START/SENDPASS/CHAP processed
00:03:23: TAC+: periodic timer stopped (queue empty)
```

The TACACS messages are intended to be self-explanatory or for consumption by service personnel only. However, the messages shown are briefly explained in the following text.

The following message indicates that a TCP open request to host 192.168.58.104 on port 1049 will time out in 15 seconds if it gets no response:

```
00:03:16: TAC+: Opening TCP/IP to 192.168.58.104/1049 timeout=15
```

The following message indicates a successful open operation and provides the address of the internal TCP “handle” for this connection:

```
00:03:16: TAC+: Opened TCP/IP handle 0x48A87C to 192.168.58.104/1049
```

The following message indicates that a TACACS+ request has been queued:

```
00:03:16: TAC+: 192.168.58.104 req=3BD868 id=-1242409656 ver=193 handle=0x48A87C (ESTAB)
expire=14 AUTHEN/START/SENDAUTH/CHAP queued
```

The message identifies the following:

- Server that the request is destined for
- Internal address of the request
- TACACS+ ID of the request
- TACACS+ version number of the request
- Internal TCP handle the request uses (which will be zero for a single-connection server)
- TCP status of the connection—which is one of the following:
  - CLOSED
  - LISTEN
  - SYNSENT
  - SYNRCVD
  - ESTAB
  - FINWAIT1
  - FINWAIT2
  - CLOSEWAIT
  - LASTACK
  - CLOSING
  - TIMEWAIT
- Number of seconds until the request times out
- Request type

The following message indicates that all 46 bytes were written to address 192.168.58.104 for request 3BD868:

```
00:03:17: TAC+: 192.168.58.104 ESTAB 3BD868 wrote 46 of 46 bytes
```

The following message indicates that 12 bytes were read in reply to the request:

```
00:03:22: TAC+: 192.168.58.104 CLOSEWAIT read=12 wanted=12 alloc=12 got=12
```

The following message indicates that 49 more bytes were read, making a total of 61 bytes in all, which is all that was expected:

```
00:03:22: TAC+: 192.168.58.104 CLOSEWAIT read=61 wanted=61 alloc=61 got=49
```

The following message indicates that a complete 61-byte reply has been read and processed for request 3BD868:

```
00:03:22: TAC+: 192.168.58.104 received 61 byte reply for 3BD868 00:03:22: TAC+:
req=3BD868 id=-1242409656 ver=193 handle=0x48A87C (CLOSEWAIT) expire=9
AUTHEN/START/SENDAUTH/CHAP processed
```

The following message indicates that the TACACS+ server helper process switched itself off when it had no more work to do:

```
00:03:22: TAC+: periodic timer stopped (queue empty)
```

Related Commands

Command	Description
debug aaa accounting	Displays information on accountable events as they occur.
debug aaa authentication	Displays information on AAA/TACACS+ authentication.
debug aaa authorization	Displays information on AAA/TACACS+ authorization.
debug sw56	Displays debug information for switched 56 K services.