

debug sntp adjust

To display information about Simple Network Time Protocol (SNTP) clock adjustments, use the **debug sntp adjust** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sntp adjust

no debug sntp adjust

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debug sntp adjust** command when an offset to the time reported by the configured NTP server is calculated. The offset indicates the difference between the router time and the actual time (as kept by the server) and is displayed in milliseconds. The clock time is then successfully changed to the accurate time by adding the offset to the current router time.

```
Router# debug sntp adjust  
  
Delay calculated, offset 3.48  
Clock slewed.
```

The following is sample output from the **debug sntp adjust** command when an offset to the time reported by a broadcast server is calculated. Because the packet is a broadcast packet, no transmission delay can be calculated. However, in this case, the offset is too large, so the clock is reset to the correct time.

```
Router# debug sntp adjust  
  
No delay calculated, offset 11.18  
Clock stepped.
```

debug snmp packets

To display information about Simple Network Time Protocol (SNTP) packets sent and received, use the **debug snmp packets** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug snmp packets

no debug snmp packets

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debug snmp packets** command when a message is received:

```
Router# debug snmp packets

Received SNMP packet from 172.16.186.66, length 48
 leap 0, mode 1, version 3, stratum 4, ppoll 1024
 rtdel 00002B00, rtdsp 00003F18, refid AC101801 (172.16.24.1)
 ref B7237786.ABF9CDE5 (23:28:06.671 UTC Tue May 13 1997)
 org 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
 rec 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
 xmt B7237B5C.A7DE94F2 (23:44:28.655 UTC Tue May 13 1997)
 inp AF3BD529.810B66BC (00:19:53.504 UTC Mon Mar 1 1993)
```

The following is sample output from the **debug snmp packets** command when a message is sent:

```
Router# debug snmp packets

Sending SNMP packet to 172.16.25.1
 xmt AF3BD455.FBBE3E64 (00:16:21.983 UTC Mon Mar 1 1993)
```

[Table 325](#) describes the significant fields shown in the display.

Table 325 *debug snmp packets Field Descriptions*

Field	Description
length	Length of the SNMP packet.
leap	Indicates if a leap second will be added or subtracted.
mode	Indicates the mode of the router relative to the server sending the packet.
version	SNTP version number of the packet.
stratum	Stratum of the server.
ppoll	Peer polling interval.
rtdel	Total delay along the path to the root clock.
rtdsp	Dispersion of the root path.
refid	Address of the server that the router is currently using for synchronization.

Table 325 *debug sntp packets Field Descriptions (continued)*

Field	Description
ref	Reference time stamp.
org	Originate time stamp. This value indicates the time the request was sent by the router.
rec	Receive time stamp. This value indicates the time the request was received by the Sntp server.
xmt	Transmit time stamp. This value indicates the time the reply was sent by the Sntp server.
inp	Destination time stamp. This value indicates the time the reply was received by the router.

debug sntp select

To display information about Simple Network Time Protocol (SNTP) server selection, use the **debug sntp select** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sntp select

no debug sntp select

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Examples The following is sample output from the **debug sntp select** command. In this example, the router will synchronize its time to the server at 172.16.186.66.

```
Router# debug sntp select  
  
SNTP: Selected 172.16.186.66
```

debug software authenticity

To debug software authenticity events, use the **debug software authenticity** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

```
debug software authenticity { envelope | errors | key | revocation | show | verbose }
```

```
no debug software authenticity { envelope | errors | key | revocation | show | verbose }
```

Syntax Description

envelope	Enables the display of all debugging output related to software authentication envelope events.
errors	Enables the display of all debugging output related to software authentication errors.
key	Enables the display of all debugging output related to software authentication key events.
revocation	Enables the display of all debugging output related to software authentication revocation events.
show	Enables the display of all debugging output related to the show software authenticity file , show software authenticity keys , and show software authenticity running commands.
verbose	Enables the display of all debugging output related to software authentication errors and events.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)M	This command was introduced for the Cisco 1941, 2900, and 3900 routers.
15.0(1)M2	This command was modified. The revocation keyword was added.
15.1(1)T	This command was integrated into Cisco IOS Release 15.1(1)T.

Usage Guidelines

Use the **debug software authenticity** command to enable debugging related to software authentication events.

Use the command in conjunction with the **show software authenticity file**, **show software authenticity keys**, **show software authenticity running**, and **show software authenticity upgrade-status** commands in order to display the debugging-related messages. For further information on these commands, see the [Cisco IOS Master Command List, All Releases](#).

Examples

The following example enables the display of debugging output related to software authentication errors:

```
Router# debug software authenticity errors
```

Software Authenticity Errors debugging is on

The following example enables the display of debugging output related to software authentication key errors, and the output from the **show software authenticity keys** command displays the key information related to software authentication debugging:

```
Router# debug software authenticity key
```

Software Authenticity Key debugging is on

```
Router# show software authenticity keys
```

Public Key #1 Information

Key Type : Release (Primary)

Public Key Algorithm : RSA

Modulus :

CC:CA:40:55:8C:71:E2:4A:3A:B6:9D:5C:94:1D:02:BA:

.....

26:04:6B:33:EB:70:2B:18:24:C7:D9:31:3E:77:24:85

Exponent : xxx

Key

*May 14 23:23:13.988: code_sign_parse_key_record: START. list offset:(0), tlv tag: 0xAE, tlv len: 281

*May 14 23:23:13.988: code_sign_parse_key_record: Tag (0xAE) found at offset: 0, list_offset: 0

*May 14 23:23:13.988: code_sign_parse_key_record: key_rec_len: 281, pub key size: 288, offset: 3

*May 14 23:23:13.988: code_sign_parse_key_record: Key Start magic: 0xxxxxxxD, at offset: 3

*May 14 23:23:13.988: code_sign_validate_key_end_magic: End Magic (0xBEEFCAFE) found at the end of the key record (292)

*May 14 23:23:13.988: code_sign_parse_key_record: Tlv start offset: 7, pub key size: 288

*May 14 23:23:13.988: code_sign_parse_key_record: Tag (Key Type:(0x1) found at offset: 7

*May 14 23:23:13.988: code_sign_parse_key_record: We increment offset by sizeof tlv: 3, size of len: 2

*May 14 23:23:13.988: code_sign_parse_key_record: Key Type: 0x1, offset: 11

*May 14 23:23:13.988: code_sign_parse_key_record: Tag (Signature Algorithm:(0x2) found at offset: 11

*May 14 23:23:13.988: code_sign_parse_key_record: We increment offset by sizeof tlv: 3, size of len: 2

*May 14 23:23:13.988: code_sign_parse_key_record: Signature Algo: 0x1, offset: 15

*May 14 23:23:13.988: code_sign_parse_key_record: Tag (Key Info Length:(0x3) found at offset: 15

*May 14 23:23:13.988: code_sign_parse_key_record: We increment offset by sizeof tlv: 3, size of len: 2

*May 14 23:23:13.988: code_sign_parse_key_record:Length (266) for type (Key Info Length), offset: 18

*May 14 23:23:13.988: code_sign_parse_key_record: Key Info Len: 266, offset: 18

*May 14 23:23:13.988: code_sign_parse_key_record: Tag (Modulus:(xxx) found at offset: 18

*May 14 23:23:13.988: code_sign_parse_key_record: We increment offset by sizeof tlv: 3, size of len: 2

*May 14 23:23:13.988: code_sign_parse_key_record: offset: 277, Modulus size: (xxx)

CCCA40558C71E24A3AB69D5C941D02BA63CDF0202FC6CBC1D73E8F27E3DA6DC615EB2FD0A66643D82BE17F3CE8
.....

47AE5135955C58B164320B925608DA4002B75FB01EFEC2691B188D6FB2E3AFE8F453888FE063B4304DDC2EB25B

*May 14 23:23:13.988: code_sign_parse_key_record: Tag (Public Exponent:(xxx) found at offset: 277

*May 14 23:23:13.988: code_sign_parse_key_record: We increment offset by sizeof tlv: 3, size of len: 2

*May 14 23:23:13.988: code_sign_parse_key_record: offset: 284, Public Exponent size: (xxx), public exponent: xxx

```
*May 14 23:23:13.988: code_sign_parse_key_record: Tag (Key Version:(0x6) found at offset:
284
*May 14 23:23:13.988: code_sign_parse_key_record: We increment offset by sizeof tlv: 3,
size of len: 2
*May 14 23:23:13.988: code_sign_parse_key_record: Key Version: 0x41, offset: 288
*May 14 23:23:13.988: code_sign_parse_key_record: END. offset (292), bitlist:
(0x3F)Version      : A
```

The following example enables the display of debugging output related to software authentication errors and events (the full range of messages), and the output from the **show software authenticity file** command displays the file information related to software authentication debugging:

```
Router# debug software authenticity verbose
```

```
Software Authenticity Verbose debugging is on
```

```
Router# show software authenticity file flash0:c3900-universalk9-mz.SSA
```

```
#####
Signature Envelope
```

```
Version 1.xxx
hdr_length xxx
signer_id_len xxx
signer_name CN=CiscoSystems;OU=C3900;O=CiscoSystems
ca_serial_num len xxx
ca_serial_num xxx
ca_name CN=CiscoSystems;OU=C3900;O=CiscoSystems
digest_algo xxx
sign_algo xxx
mod_size xxx
key_type xxx
key_version 0xx1
signature length xxx
signature TLV offset xxx
signature
4F94AC7EAA7B9B9EAE66EFA8BF426C3BFE622D7C651A35F686F7DD7FBF329317B269CAEADB5679834B93BF2C91
.....
F160EF79B82AB41176975D024D1DA9EB75499BC139BFED9AF8D3F4DFAE35BFC0CDA1519F7CD9C8EB08D8D09D18
--More--
*May 28 08:05:44.487: code_sign_get_image_type: filename:flash0:c3900-universalk9-mz.SSA
*May 28 08:05:44.487: cs_open: Opened file flash0:c3900-universalk9-mz.SSA with fd=13
*May 28 08:05:44.491: code_sign_get_image_type: image type found: image (elf) (3)
*May 28 08:05:44.491: code_sign_get_image_envelope Start, fd(13)
*May 28 08:05:44.491: code_sign_get_number_of_sections num_sections: 7

*May 28 08:05:44.547: code_sign_get_image_envelope:SHA2 Note Section found at iter: 6
*May 28 08:05:44.547: code_sign_get_image_envelope: Note name len(n_namesz): 13, Signature
Env Len(n_descz): 388

*May 28 08:05:44.547: code_sign_get_image_envelope: sizeof elf_note_hdr: 12, size of
Elf32_Nhdr: 12
*May 28 08:05:44.547: code_sign_get_image_envelope: Note Name: (CISCO SYSTEMS) fo
#####
```

```
File Name      : flash0:c3900-universalk9-mz.SSA
Image type     : Development
  Signer Information
    Common Name      : xxx
    Organization Unit : xxx
    Organization Name : xxx
  Certificate Serial Number : xxx
  Hash Algorithm     : SHA512
  Signature Algorithm : 2048-bit RSA
  Key Version        : A
```

Related Commands

Command	Description
show software authenticity file	Displays information related to software authentication for the loaded image file.
show software authenticity keys	Displays the software public keys that are in the storage with the key types.
show software authenticity running	Displays software authenticity information for the current ROMmon and Cisco IOS image used for booting.
show software authenticity upgrade-status	Displays software authenticity information indicating if the digitally signed software has been signed with a new production key after a production key revocation.

debug source bridge

To display information about packets and frames transferred across a source-route bridge, use the **debug source bridge** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug source bridge

no debug source bridge

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debug source bridge** command for peer bridges using TCP as a transport mechanism. The remote source-route bridging (RSRB) network configuration has ring 2 and ring 1 bridged together through remote peer bridges. The remote peer bridges are connected via a serial line and use TCP as the transport mechanism.

```
Router# debug source bridge

RSRB: remote explorer to 5/192.108.250.1/1996 srn 2 [C840.0021.0050.0000]
RSRB: Version/Ring XReq sent to peer 5/192.108.250.1/1996
RSRB: Received version reply from 5/192.108.250.1/1996 (version 2)
RSRB: DATA: 5/192.108.250.1/1996 Ring Xchg Rep, trn 2, vrn 5, off 18, len 10
RSRB: added bridge 1, ring 1 for 5/192.108.240.1/1996
RSRB: DATA: 5/192.108.250.1/1996 Explorer trn 2, vrn 5, off 18, len 69
RSRB: DATA: 5/192.108.250.1/1996 Forward trn 2, vrn 5, off 0, len 92
RSRB: DATA: forward Forward srn 2, br 1, vrn 5 to peer 5/192.108.250.1/1996
```

The following line indicates that a remote explorer frame has been sent to IP address 192.108.250.1 and, like all RSRB TCP connections, has been assigned port 1996. The bridge belongs to ring group 5. The explorer frame originated from ring 2. The routing information field (RIF) descriptor has been generated by the local station and indicates that the frame was sent out via bridge 1 onto virtual ring 5.

```
RSRB: remote explorer to 5/192.108.250.1/1996 srn 2 [C840.0021.0050.0000]
```

The following line indicates that a request for remote peer information has been sent to IP address 192.108.250.1, TCP port 1996. The bridge belongs to ring group 5.

```
RSRB: Version/Ring XReq sent to peer 5/192.108.250.1/1996
```

The following line is the response to the version request previously sent. The response is sent from IP address 192.108.250.1, TCP port 1996. The bridge belongs to ring group 5.

```
RSRB: Received version reply from 5/192.108.250.1/1996 (version 2)
```

The following line is the response to the ring request previously sent. The response is sent from IP address 192.108.250.1, TCP port 1996. The target ring number is 2, virtual ring number is 5, the offset is 18, and the length of the frame is 10 bytes.

```
RSRB: DATA: 5/192.108.250.1/1996 Ring Xchg Rep, trn 2, vrn 5, off 0, len 10
```

The following line indicates that bridge 1 and ring 1 were added to the source-bridge table for IP address 192.108.250.1, TCP port 1996:

```
RSRB: added bridge 1, ring 1 for 5/192.108.250.1/1996
```

The following line indicates that a packet containing an explorer frame came across virtual ring 5 from IP address 192.108.250.1, TCP port 1996. The packet is 69 bytes in length. This packet is received after the Ring Exchange information was received and updated on both sides.

```
RSRB: DATA: 5/192.108.250.1/1996 Explorer trn 2, vrn 5, off 18, len 69
```

The following line indicates that a packet containing data came across virtual ring 5 from IP address 192.108.250.1 over TCP port 1996. The packet is being placed on the local target ring 2. The packet is 92 bytes in length.

```
RSRB: DATA: 5/192.108.250.1/1996 Forward trn 2, vrn 5, off 0, len 92
```

The following line indicates that a packet containing data is being forwarded to the peer that has IP address 192.108.250.1 address belonging to local ring 2 and bridge 1. The packet is forwarded via virtual ring 5. This packet is sent after the Ring Exchange information was received and updated on both sides.

```
RSRB: DATA: forward Forward srn 2, br 1, vrn 5 to peer 5/192.108.250.1/1996
```

The following is sample output from the **debug source bridge** command for peer bridges using direct encapsulation as a transport mechanism. The RSRB network configuration has ring 1 and ring 2 bridged together through peer bridges. The peer bridges are connected via a serial line and use TCP as the transport mechanism.

```
Router# debug source bridge
```

```
RSRB: remote explorer to 5/Serial1 srn 1 [C840.0011.0050.0000]
RSRB: Version/Ring XReq sent to peer 5/Serial1
RSRB: Received version reply from 5/Serial1 (version 2)
RSRB: IFin: 5/Serial1 Ring Xchg, Rep trn 0, vrn 5, off 0, len 10
RSRB: added bridge 1, ring 1 for 5/Serial1
```

The following line indicates that a remote explorer frame was sent to remote peer Serial1, which belongs to ring group 5. The explorer frame originated from ring 1. The RIF descriptor 0011.0050 was generated by the local station and indicates that the frame was sent out via bridge 1 onto virtual ring 5.

```
RSRB: remote explorer to 5/Serial1 srn 1 [C840.0011.0050.0000]
```

The following line indicates that a request for remote peer information was sent to Serial1. The bridge belongs to ring group 5.

```
RSRB: Version/Ring XReq sent to peer 5/Serial1
```

The following line is the response to the version request previously sent. The response is sent from Serial 1. The bridge belongs to ring group 5 and the version is 2.

```
RSRB: Received version reply from 5/Serial1 (version 2)
```

The following line is the response to the ring request previously sent. The response is sent from Serial1. The target ring number is 2, virtual ring number is 5, the offset is 0, and the length of the frame is 39 bytes.

```
RSRB: IFin: 5/Serial1 Ring Xchg Rep, trn 2, vrn 5, off 0, len 39
```

The following line indicates that bridge 1 and ring 1 were added to the source-bridge table for Serial1:

```
RSRB: added bridge 1, ring 1 for 5/Serial1
```

debug source error

To display source-route bridging (SRB) errors, use the **debug source error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug source error

no debug source error

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The debug source error command displays some output also found in the **debug source bridge** output. See the **debug source bridge** command for other possible output.

Examples

In all of the following examples of **debug source error** command messages, the variable *number* is the Token Ring interface. For example, if the line of output starts with SRB1, the output relates to the Token Ring 1 interface. SRB indicates a source-route bridging message. RSRB indicates a remote source-route bridging message. SRTLTLB indicates a source-route translational bridging (SR/TLB) message.

In the following example, a packet of protocol *protocol-type* was dropped:

```
SRBnumber drop: Routed protocol protocol-type
```

In the following example, an Address Resolution Protocol (ARP) packet was dropped. ARP is defined in RFC 826.

```
SRBnumber drop:TYPE_RFC826_ARP
```

In the following example, the current Cisco IOS version does not support Qualified Logical Link Control (QLLC). Reconfigure the router with an image that has the IBM feature set.

```
RSRB: QLLC not supported in version version  
Please reconfigure.
```

In the following example, the packet was dropped because the outgoing interface of the router was down:

```
RSRB IF: outgoing interface not up, dropping packet
```

In the following example, the router received an out-of-sequence IP sequence number in a Fast Sequenced Transport (FST) packet. FST has no recovery for this problem like TCP encapsulation does.

```
RSRB FST: bad sequence number dropping.
```

In the following example, the router was unable to locate the virtual interface:

```
RSRB: couldn't find virtual interface
```

In the following example, the TCP queue of the peer router is full. TCPD indicates that this is a TCP debug.

```
RSRB TCPD: tcp queue full for peer
```

In the following example, the router was unable to send data to the *peer* router. A *result* of 1 indicates that the TCP queue is full. A *result* of —1 indicates that the RSRB peer is closed.

```
RSRB TCPD: tcp send failed for peer result
```

In the following example, the routing information identifier (RII) was not set in the explorer packet going forward. The packet will not support SRB, so it is dropped.

```
vrforward_explorer - RII not set
```

In the following example, a packet sent to a virtual bridge in the router did not include a routing information field (RIF) to tell the router which route to use:

```
RSRB: no RIF on packet sent to virtual bridge
```

The following example indicates that the RIF did not contain any information or the length field was set to zero:

```
RSRB: RIF length of zero sent to virtual bridge
```

The following message occurs when the local service access point (LSAP) is out of range. The variable *lsap-out* is the value, *type* is the type of RSRB peer, and *state* is the state of the RSRB peer.

```
VRP: rsrb_lsap_out = lsap-out, type = type, state = state
```

In the following message, the router is unable to find another router with which to exchange bridge protocol data units (BPDUs). BPDUs are exchanged to set up the spanning tree and determine the forwarding path.

```
RSRB(span): BPDUs peer not found
```

Related Commands

Command	Description
debug source bridge	Displays information about packets and frames transferred across a source-route bridge.
debug source event	Displays information on SRB activity.

debug source event

To display information on source-route bridging (SRB) 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 326](#) describes the significant fields shown in the display.

Table 326 *debug source event Field Descriptions*

Field	Description
RSRB0:	Indication that this routing information field (RIF) cache entry is for the Token Ring interface 0, which has been configured for remote source-route bridging (SRB). (SRB 1, 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 326 *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:

```
RSRBN: 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 bridge protocol data unit (BPDU) packet:

```
Router# debug span
```

```
ST: Ether4 00000000000000A080002A02D6700000000000A080002A02D6780010000140002000F00
```

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

```
ST: Ether4 00000000000000A080002A02D6700000000000A080002A02D6780010000140002000F00
A  B C D E  F           G           H  I           J K L  M  N  O
```

[Table 327](#) describes the significant fields shown in the display.

Table 327 *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: <ul style="list-style-type: none"> 00 indicates config BPDU. 80 indicates the Topology Change Notification (TCN) BPDU.
(D) 00	Topology change acknowledgment: <ul style="list-style-type: none"> 00 indicates no change. 80 indicates a change notification.
(E) 000A	Root priority.

Table 327 *debug span Field Descriptions—IEEE BPDU Packet (continued)*

Field	Description
(F) 080002A02D67	Root ID.
(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 328](#) describes the significant fields shown in the display.

Table 328 *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: <ul style="list-style-type: none"> • 0x19—DEC Hello • 0x02—TCN
(C) 01	DEC version.
(D) 00	Flag that is a bit field with the following mapping: <ul style="list-style-type: none"> • 1—TCN • 2—TCN acknowledgment • 8—Use short timers
(E) 0002	Root priority.
(F) 00000C01A2C9	Root ID (MAC address).

Table 328 *debug span Field Descriptions for a DEC BPDU Packet (continued)*

Field	Description
(G) 0064	Root path cost (translated as 100 in decimal notation).
(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 spanning-tree

To debug spanning-tree activities, use the **debug spanning-tree** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug spanning-tree {all | backbonefast | bpdud | bpdud-opt | config | etherchannel | events |
exceptions | general | pvst+ | root | snmp | uplinkfast}
```

```
no debug spanning-tree {all | backbonefast | bpdud | bpdud-opt | config | etherchannel | events |
exceptions | general | pvst+ | root | snmp | uplinkfast}
```

Syntax Description

all	Displays all spanning-tree debugging messages.
backbonefast	Displays debugging messages for BackboneFast events.
bpdud	Displays debugging messages for spanning-tree Bridge Protocol Data Units (BPDUs).
bpdud-opt	Displays debugging messages for optimized BPDU handling.
config	Displays debugging messages for spanning-tree configuration changes.
etherchannel	Displays debugging messages for EtherChannel support.
events	Displays debugging messages for spanning-tree topology events.
exceptions	Displays debugging messages for spanning-tree exceptions.
general	Displays debugging messages for general spanning-tree activity.
pvst+	Displays debugging messages for per-VLAN Spanning Tree Plus (PVST+) events.
root	Displays debugging messages for spanning-tree root events.
snmp	Displays debugging messages for spanning-tree Simple Network Management Protocol (SNMP) handling.
uplinkfast	Displays debugging messages for UplinkFast events.

Defaults

Debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Release	Modification
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command is supported only by the Supervisor Engine and can be entered only from the switch console.

The **undebg spanning-tree** command is the same as the **no debug spanning-tree** command.

Related Commands

Command	Description
show debugging	Displays information about the types of debugging that are enabled.
show spanning-tree	Displays spanning-tree state information.

debug ss7 mtp1


Note

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

To initiate Signaling System 7 (SS7) Message Transfer Part Level 1 (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	(Optional) Initiates SS7 MTP2 debugging.
ipc	(Optional) Initiates SS7 MTP1 debugging for HOST/FW IPC.
link-state	(Optional) Initiates SS7 MTP1 debugging for link-state transitions.
oir	(Optional) Initiates SS7 MTP1 trunk dial feature card (DFC) online insertion and removal (OIR) debugging.
rx	(Optional) Initiates SS7 MTP1 debugging for receive events. Not used in Release 12.2(11)T.
scc-regs	(Optional) Initiates SS7 MTP1 debugging for SCC registers. Not used in Release 12.2(11)T.
siram	(Optional) Initiates SS7 MTP1 debugging for siram values. Not used in Release 12.2(11)T.
tdm-info	(Optional) Initiates SS7 MTP1 debugging for time-division multiplexing (TDM) information.
tx	(Optional) Initiates SS7 MTP1 debugging for transmission events. Not used in Release 12.2(11)T.

Defaults

Debug is disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.2(11)T	This command was introduced on the Cisco AS5350 and Cisco AS5400 Signaling Link Terminal (SLT).
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

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 = 3 should be *->channel.

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

Related Commands

Command	Description
debug ss7 sm	Displays debugging messages for an SS7 Session Manager.

debug ss7 mtp2

To trace backhaul Signaling System 7 (SS7) Message Transfer Part Level 2 (MTP2) 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 | suerm | timer |
txc][channel]
```

```
no debug ss7 mtp2
```

Syntax Description

aerm	(Optional) Initiates alignment Error Rate Monitor events.
backhaul	(Optional) Initiates trace backhaul control messages. The <i>channel</i> argument represents a logical channel number. Valid values are from 0 to 3.
cong	(Optional) Initiates congestion Control events.
iac	(Optional) Initiates initial Alignment Control events.
lsc	(Optional) Initiates Link State Control events.
lssu	(Optional) Initiates trace backhaul LSSU messages.
msu	(Optional) Initiates trace backhaul MSU messages (use during low traffic only).
packet [all]	(Optional) 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	(Optional) Displays information about SS7 MTP2 receiver state machine events and transitions.
suerm	(Optional) Displays information about SS7 MTP2 Signal Unit Error Rate Monitor (SUERM) state machine events and transitions.
timer	(Optional) Displays information about SS7 MTP2 timer starts and stops.
txc	(Optional) Displays information about SS7 MTP2 transmit state machine events and transitions.
<i>channel</i>	(Optional) The channel argument represents a logical channel number. Valid values are from 0 to 3.

Defaults

Debug is disabled.

Command Modes

Global configuration

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 implemented on the Cisco AS5350 and Cisco AS5400 Cisco Signaling Link Terminal (SLT).
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

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

Examples

The following is sample output from the **debug ss7 mtp2 aerm** command. See the MTP2 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 MTP2 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 MTP2 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 MTP2 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 MTP2 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
```



Caution

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 MTP2 specification tables for details:

```

Router# debug ss7 mtp2 rcv 0

*Mar 8 09:22:35.160:itu2RC_Stop chnl=0 MTP2RC_INSERTSERVICE
*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 MTP2 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
```

**Caution**

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
```

**Caution**

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 MTP2 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

Backhaul Debug Reason Codes	Description
0x6	Two or three invalid FIBs
0x7	LSSU (Link Status Signal Unit) condition
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 ss7 sm	Displays debugging messages for an SS7 Session Manager.

debug ss7 sm

To display debugging messages for an 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 session-id | set | timer]
```

```
no debug ss7 sm session
```

Syntax Description

session	(Optional) Sets Session Manager session debug.
<i>session-id</i>	(Optional) Specifies a session ID number from 0 to 3.
set	(Optional) Sets Session Manager debug.
timer	(Optional) Sets Session Manager timer debug.

Defaults

Debug is disabled.

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

Usage Guidelines

Use this command to 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 to 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:rdp 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:rdp signal RUDP_SOFT_RESET_SIG, session = 0
*Mar  8 09:37:58.129:SM:rdp 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:rdp 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.

Defaults 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.
	12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
	12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines Use this command to 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.

Defaults

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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command 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 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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to 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 ctrl-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	Enables NRP-SSG to forward packets locally.

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 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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

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 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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

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 dhcp

To enable the display of control errors and events related to Service Selection Gateway (SSG) Dynamic Host Configuration Protocol (DHCP), use the **debug ssg dhcp** command in **privileged EXEC** mode. To stop debugging, use the **no** form of this command.

```
debug ssg dhcp {error | event} [ip-address]
```

```
no debug ssg dhcp {error | event} [ip-address]
```

Syntax Description

error	Enables the display of SSG-DHCP control error information.
event	Enables the display of SSG-DHCP control events information.
<i>ip-address</i>	(Optional) Limits the display of information to the specified IP address.

Command Default

Displays SSG-DHCP information for all IP addresses.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

SSG DHCP Event Messages

The following example shows user login events when DHCP intercept is enabled using the **ssg intercept dhcp** command.

```
debug ssg dhcp
```

```
01:01:03: DHCPD: remote id 020a000005010101100000000000
01:01:03: DHCPD: circuit id 00000000
01:01:03: SSG-DHCP-EVN: DHCP-DISCOVER event received. SSG-dhcp awareness feature enabled
01:01:03: DHCPD: DHCPDISCOVER received from client
0063.6973.636f.2d30.3030.632e.3331.6561.2e61.3963.312d.4661.302f.31 on interface
FastEthernet1/0.
01:01:03: DHCPD: Seeing if there is an internally specified pool class:
01:01:03: DHCPD: htype 1 chaddr 000c.31ea.a9c1
01:01:03: DHCPD: remote id 020a000005010101100000000000
01:01:03: DHCPD: circuit id 00000000
01:01:03: SSG-DHCP-EVN: Get pool name called for 000c.31ea.a9c1. No hostobject
01:01:03: SSG-DHCP-EVN: Get pool class called, class name =
01:01:03: DHCPD: No internally specified class returned
01:01:03: DHCPD: Sending DHCPPOFFER to client
0063.6973.636f.2d30.3030.632e.3331.6561.2e61.3963.312d.4661.302f.31 (5.1.1.2).
01:01:03: DHCPD: child pool: 5.1.1.0 / 255.255.255.0 (Default-pool)
01:01:03: DHCPD: pool Default-pool has no parent.
```

```

01:01:03: DHCPD: child pool: 5.1.1.0 / 255.255.255.0 (Default-pool)
01:01:03: DHCPD: pool Default-pool has no parent.
01:01:03: DHCPD: child pool: 5.1.1.0 / 255.255.255.0 (Default-pool)
01:01:03: DHCPD: pool Default-pool has no parent.
01:01:03: DHCPD: broadcasting BOOTREPLY to client 000c.31ea.a9c1.
01:01:03: DHCPD: DHCPREQUEST received from client
0063.6973.636f.2d30.3030.632e.3331.6561.2e61.3963.312d.4661.302f.31.
01:01:03: DHCPD: Sending notification of ASSIGNMENT:
01:01:03: DHCPD: address 5.1.1.2 mask 255.255.255.0
01:01:03: DHCPD: htype 1 chaddr 000c.31ea.a9c1
01:01:03: DHCPD: lease time remaining (secs) = 180
01:01:03: SSG-DHCP-EVN:5.1.1.2: IP address notification received.
01:01:03: SSG-DHCP-EVN:5.1.1.2: HostObject not present
01:01:03: DHCPD: No default domain to append - abort update
01:01:03: DHCPD: Sending DHCPACK to client
0063.6973.636f.2d30.3030.632e.3331.6561.2e61.3963.312d.4661.302f.31 (5.1.1.2).
01:01:03: DHCPD: child pool: 5.1.1.0 / 255.255.255.0 (Default-pool)
01:01:03: DHCPD: pool Default-pool has no parent.
01:01:03: DHCPD: child pool: 5.1.1.0 / 255.255.255.0 (Default-pool)
01:01:03: DHCPD: pool Default-pool has no parent.
01:01:03: DHCPD: child pool: 5.1.1.0 / 255.255.255.0 (Default-pool)
01:01:03: DHCPD: pool Default-pool has no parent.
01:01:03: DHCPD: broadcasting BOOTREPLY to client 000c.31ea.a9c1.

```

SSG DHCP Error Messages

The following example shows user login errors when a user tries to log into two different services that require IP addresses to be assigned from different pools.

debug ssg dhcp error

```

01:21:58: SSG-CTL-EVN: Checking maximum service count.
01:21:58: SSG-CTL-EVN: Service logon is accepted.
01:21:58: SSG-CTL-EVN: Activating the ConnectionObject.

01:21:58: SSG-DHCP-ERR:6.2.1.2: DHCP pool name of this service is different from, users
already logged in service DHCP pool name
01:21:58: SSG-CTL-EVN: Connection Activation Failed for host 6.2.1.2

01:21:58: SSG-CTL-EVN: Send cmd 11 to host S6.2.1.2. dst=10.76.86.90:42412
01:21:58: SSG-CTL-PAK: Sent packet:
01:21:58: RADIUS: id= 0, code= Access-Reject, len= 79

```

Related Commands

Command	Description
<code>ssg intercept dhcp</code>	Configures SSG to assign IP addresses from a user's ISP.

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 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.
	12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
	12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines 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

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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

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

debug ssg packets



Note

Effective with Release 12.2(13)T, the **debug ssg packets** command is replaced by the **debug ssg tcp-redirect** command. See the **debug ssg tcp-redirect** command for more information.

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

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.
12.2(13)T	This command was replaced by the debug ssg tcp-redirect command.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

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)
```

■ 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 (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 debugging 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.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)B	This command was introduced on the Cisco 6400 series.
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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command displays debugging 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 events

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
```

debug ssg port-map

```
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.
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. This command replaces the debug ssg packets command.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to 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.

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.

Command	Description
<code>ssg enable</code>	Enables SSG.
<code>ssg tcp-redirect</code>	Enables SSG TCP redirect and enters SSG-redirect mode.

debug ssg transparent login

To display all the Service Selection Gateway (SSG) transparent login control events or errors, use the **debug ssg transparent login** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ssg transparent login {errors | events} [ip-address]
```

```
no debug ssg transparent login {errors | events} [ip-address]
```

Syntax Description

errors	Displays any SSG transparent login errors.
events	Displays significant SSG transparent login events or state changes.
<i>ip-address</i>	(Optional) Displays events or errors for a specified IP address.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(1a)BW	This command was introduced.
12.3(3)B	This command was integrated into Cisco IOS Release 12.3(3)B.
12.3(7)T	This command was integrated into Cisco IOS Release 12.3(7)T.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command when troubleshooting SSG for problems related to transparent autologon users.

Examples

The following examples show sample output from the **debug ssg transparent login** command. The output is self-explanatory.

Unidentified (NR) User Example

```
*Jan 15 12:34:47.847:SSG-TAL-EVN:100.0.0.2 :Added entry successfully
*Jan 15 12:34:47.847:SSG-TAL-EVN:100.0.0.2 :Attempting authorization
*Jan 15 12:34:47.847:SSG-TAL-EVN:100.0.0.2 :Attempting to send authorization request
*Jan 15 12:35:09.711:SSG-TAL-EVN:100.0.0.2 :Authorization response received
*Jan 15 12:35:09.711:SSG-TAL-EVN:100.0.0.2 :Authorization timedout. User statechanged to
unidentified
*Jan 15 12:35:09.711:%SSG-5-SSG_TAL_NR:SSG TAL :No response from AAA server. AAA server
might be down or overloaded.
*Jan 15 12:35:09.711:SSG-TAL-EVN:100.0.0.2 :Start SP/NR entry timeout timer for 10 mins
```

Transparent Pass-Through (TP) User Example

```
*Jan 15 12:40:39.875:SSG-TAL-EVN:100.0.0.2 :Added entry successfully
*Jan 15 12:40:39.875:SSG-TAL-EVN:100.0.0.2 :Attempting authorization
*Jan 15 12:40:39.875:SSG-TAL-EVN:100.0.0.2 :Attempting to send authorization request
*Jan 15 12:40:39.879:SSG-TAL-EVN:100.0.0.2 :Authorization response received
```

```
*Jan 15 12:40:39.879:SSG-TAL-EVN:100.0.0.2 :Parsing profile for TP attribute
*Jan 15 12:40:39.879:SSG-TAL-EVN:100.0.0.2 :TP attribute found - Transparent user
*Jan 15 12:40:39.879:SSG-TAL-EVN:100.0.0.2 :Stop SP/NR timer
*Jan 15 12:40:39.879:SSG-TAL-EVN:100.0.0.2 :Idle timer started for 0 secs
*Jan 15 12:40:39.879:SSG-TAL-EVN:100.0.0.2 :Session timer started for 0 secs
```

Suspect User (SP) Example

```
*Jan 15 12:43:25.363:SSG-TAL-EVN:10.10.10.10 :Added entry successfully
*Jan 15 12:43:25.363:SSG-TAL-EVN:10.10.10.10 :Attempting authorization
*Jan 15 12:43:25.363:SSG-TAL-EVN:10.10.10.10 :Attempting to send authorization request
*Jan 15 12:43:25.939:SSG-TAL-EVN:10.10.10.10 :Authorization response received
*Jan 15 12:43:25.939:SSG-TAL-EVN:10.10.10.10 :Access reject from AAA server. Userstate
changed to suspect
*Jan 15 12:43:25.939:SSG-TAL-EVN:10.10.10.10 :Start SP/NR entry timeout timer for 60 mins
```

Clear All Users Example

The following is sample output for the **debug ssg transparent login** command when used after all transparent autologon users have been cleared by using the **clear ssg user transparent all** command.

```
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.10.10.10 :Entry removed
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.10.10.10 :Stop SP/NR timer
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.10.10.10 :Stop Idle timer
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.10.10.10 :Stop session timer
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.11.11.11 :Entry removed
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.11.11.11 :Stop SP/NR timer
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.11.11.11 :Stop Idle timer
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.11.11.11 :Stop session timer
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.0.0.2 :Entry removed
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.0.0.2 :Stop SP/NR timer
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.0.0.2 :Stop Idle timer
*Jan 15 12:47:08.943:SSG-TAL-EVN:10.0.0.2 :Stop session timer
```

Related Commands

Command	Description
ssg login transparent	Enables the SSG Transparent Autologon feature.

debug ssl

To display information about Secure Socket Layer (SSL) and Transport Layer Security (TLS) applications, use the **debug ssl** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

```
debug ssl {error | event | hdshake | traffic | openssl {errors | msg | states}}
```

```
no debug ssl {error | event | hdshake | traffic | openssl {errors | msg | states}}
```

Syntax Description

error	Displays any errors during control (negotiation) and data phases.
event	Displays SSL negotiation events.
hdshake	Displays SSL HandShake protocol information.
traffic	Displays SSL traffic messages.
openssl	Displays TLS/SSL debugging of the OpenSSL toolkit.
errors	Displays protocol errors, such as a bad packet or authentication failure.
msg	Displays hex dumps of the protocol packets.
states	Displays protocol state transitions.

Command Default

Debugging is not turned on.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.4(6)T	The openssl keyword was added.
12.4(22)T	The error , event , hdshake , and traffic keywords were removed.

Usage Guidelines

To display information about SSL and TLS applications, you should first try the **debug ssl openssl errors** command because it will display any obvious failures that are reported by the protocol layer. Next, try the **debug ssl openssl states** command to display problems that are caused by system flow issues that do not produce an error message. If you need more information, you should try the **debug ssl openssl msg** command. This output will be verbose and is rarely useful, but in some circumstances, it can provide a binary dump of the protocol packets. If the problem requires debugging at the level of the packet dumps, it is usually better to use a protocol analyzer (for example, Wireshark).



Note

The options available for the **debug ssl** command depend on the version of Cisco IOS software release. See the Command History table for the supported Cisco IOS software releases.

**Note**

It is suggested that when setting debugging, you first enable the **debug ssl openssl errors** command, **debug ssl openssl states** command, and a subset of one of the **debug crypto pki** commands. If you still do not see the problem, you might use a protocol analyzer. The **debug ssl openssl msg** command should probably be used only if you cannot get a packet trace off the wire or if you suspect that the problem is between the wire and the protocol stack.

Examples

The following example shows that the **debug ssl openssl errors** command has been configured:

```
Router# debug ssl openssl errors
```

Related Commands

Command	Description
debug crypto pki messages	Displays debugging messages for the details of the interaction (message dump) between the CA and the router.
debug crypto pki server	Enables debugging for a crypto PKI certificate server.
debug crypto pki transactions	Displays debugging messages for the trace of interaction (message type) between the CA and the router.

debug ssl openssl

To display information about Secure Socket Layer (SSL) and Transport Layer Security (TLS) applications, use the **debug ssl openssl** command in privileged EXEC mode. To turn off debugging, use the **no** form of the command.

```
debug ssl openssl {errors | msg | states}
```

```
no debug ssl openssl {errors | msg | states}
```

Syntax Description

errors	Displays protocol errors, such as a bad packet or authentication failure.
msg	Displays hex dumps of the protocol packets.
states	Displays protocol state transitions.

Command Default

Debugging is not turned on.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(22)T	This command was introduced.

Usage Guidelines

To display information about SSL and TLS applications, you must use the **debug ssl openssl errors** command, because it will display any obvious failures that are reported by the protocol layer. Next, you must use the **debug ssl openssl states** command to display problems that are caused by system flow issues that do not produce an error message. If you need more information, you must use the **debug ssl openssl msg** command. This output will be verbose and is rarely useful, but in some circumstances, it can provide a binary dump of the protocol packets. If the problem requires debugging at the level of the packet dumps, it is usually recommended to use a protocol analyzer (for example, Wireshark).

Examples

The following example shows how to enable the **debug ssl openssl errors** command :

```
Router# debug ssl openssl errors
TLS errors debugging is on
```

Related Commands

Command	Description
debug crypto pki messages	Displays debugging messages for the details of the interaction (message dump) between the CA and the router.
debug crypto pki server	Enables debugging for a crypto PKI certificate server.
debug crypto pki transactions	Displays debugging messages for the trace of interaction (message type) between the CA and the router.

debug ssm

To display diagnostic information about the Segment Switching Manager (SSM) for switched Layer 2 segments, use the **debug ssm** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

```
debug ssm { cm errors | cm events | fhm errors | fhm events | sm errors | sm events | sm counters
            | xdr }
```

```
no debug ssm { cm errors | cm events | fhm errors | fhm events | sm errors | sm events | sm
              counters | xdr }
```

Syntax Description

cm errors	Displays Connection Manager (CM) errors.
cm events	Displays CM events.
fhm errors	Displays Feature Handler Manager (FHM) errors.
fhm events	Displays FHM events.
sm errors	Displays Segment Handler Manager (SM) errors.
sm events	Displays SM events.
sm counters	Displays SM counters.
xdr	Displays external data representation (XDR) messages related to traffic sent across the backplane between Router Processors and line cards.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.2(25)S	This command was integrated to Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

The SSM manages the data-plane component of the Layer 2 Virtual Private Network (L2VPN) configuration. The CM tracks the connection-level errors and events that occur on an xconnect. The SM tracks the per-segment events and errors on the xconnect.

Use the **debug ssm** command to troubleshoot problems in bringing up the data plane.

This command is generally used only by Cisco engineers for internal debugging of SSM processes.

Examples

The following example shows sample output for the **debug ssm xdr** command:

```
Router# debug ssm xdr

SSM xdr debugging is on

2w5d: SSM XDR: [4096] deallocate segment, len 16
2w5d: SSM XDR: [8193] deallocate segment, len 16
2w5d: %LINK-3-UPDOWN: Interface FastEthernet2/1, changed state to down
2w5d: %LINK-3-UPDOWN: Interface FastEthernet2/1, changed state to up
2w5d: SSM XDR: [4102] provision segment, switch 4101, len 106
2w5d: SSM XDR: [4102] update segment status, len 17
2w5d: SSM XDR: [8199] provision segment, switch 4101, len 206
2w5d: SSM XDR: [4102] update segment status, len 17
2w5d: %SYS-5-CONFIG_I: Configured from console by console
2w5d: %LINK-3-UPDOWN: Interface FastEthernet2/1, changed state to down
2w5d: SSM XDR: [4102] update segment status, len 17
2w5d: %LINK-3-UPDOWN: Interface FastEthernet2/1, changed state to up
2w5d: SSM XDR: [4102] deallocate segment, len 16
2w5d: SSM XDR: [8199] deallocate segment, len 16
2w5d: SSM XDR: [4104] provision segment, switch 4102, len 106
2w5d: SSM XDR: [4104] update segment status, len 17
2w5d: SSM XDR: [8201] provision segment, switch 4102, len 206
2w5d: SSM XDR: [4104] update segment status, len 17
2w5d: SSM XDR: [4104] update segment status, len 17
2w5d: %SYS-5-CONFIG_I: Configured from console by console
```

The following example shows the events that occur on the segment manager when an Any Transport over MPLS (AToM) virtual circuit (VC) configured for Ethernet over MPLS is shut down and then enabled:

```
Router# debug ssm sm events

SSM Connection Manager events debugging is on

Router(config)# interface fastethernet 0/1/0.1
Router(config-subif)# shutdown

09:13:38.159: SSM SM: [SSS:AToM:36928] event Unprovison segment
09:13:38.159: SSM SM: [SSS:Ethernet Vlan:4146] event Unbind segment
09:13:38.159: SSM SM: [SSS:AToM:36928] free segment class
09:13:38.159: SSM SM: [SSS:AToM:36928] free segment
09:13:38.159: SSM SM: [SSS:AToM:36928] event Free segment
09:13:38.159: SSM SM: last segment class freed
09:13:38.159: SSM SM: [SSS:Ethernet Vlan:4146] segment ready
09:13:38.159: SSM SM: [SSS:Ethernet Vlan:4146] event Found segment data

Router(config-subif)# no shutdown

09:13:45.815: SSM SM: [SSS:AToM:36929] event Provison segment
09:13:45.815: label_oce_get_label_bundle: flags 14 label 16
09:13:45.815: SSM SM: [SSS:AToM:36929] segment ready
09:13:45.815: SSM SM: [SSS:AToM:36929] event Found segment data
09:13:45.815: SSM SM: [SSS:AToM:36929] event Bind segment
09:13:45.815: SSM SM: [SSS:Ethernet Vlan:4146] event Bind segment
```

The following example shows the events that occur on the CM when an AToM VC configured for Ethernet over MPLS is shut down and then enabled:

```
Router(config)# interface fastethernet 0/1/0.1
Router(config-subif)# shutdown

09:17:20.179: SSM CM: [AToM] unprovision segment, id 36929
09:17:20.179: SSM CM: CM FSM: state Open - event Free segment
09:17:20.179: SSM CM: [SSS:AToM:36929] unprovision segment 1
```

```
09:17:20.179: SSM CM: [SSS:AToM] shQ request send unprovision complete event
09:17:20.179: SSM CM: [SSS:Ethernet Vlan:4146] unbind segment 2
09:17:20.179: SSM CM: [SSS:Ethernet Vlan] shQ request send ready event
09:17:20.179: SSM CM: SM msg event send unprovision complete event
09:17:20.179: SSM CM: SM msg event send ready event
```

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

```
09:17:35.879: SSM CM: Query AToM to Ethernet Vlan switching, enabled
09:17:35.879: SSM CM: [AToM] provision second segment, id 36930
09:17:35.879: SSM CM: CM FSM: state Down - event Provision segment
09:17:35.879: SSM CM: [SSS:AToM:36930] provision segment 2
09:17:35.879: SSM CM: [AToM] send client event 6, id 36930
09:17:35.879: SSM CM: [SSS:AToM] shQ request send ready event
09:17:35.883: SSM CM: SM msg event send ready event
09:17:35.883: SSM CM: [AToM] send client event 3, id 36930
```

The following example shows the events that occur on the CM and SM when an AToM VC is provisioned and then unprovisioned:

```
Router# debug ssm cm events
```

```
SSM Connection Manager events debugging is on
```

```
Router# debug ssm sm events
```

```
SSM Segment Manager events debugging is on
```

```
Router# configure terminal
```

```
Router(config)# interface ethernet1/0
```

```
Router(config-if)# xconnect 10.55.55.2 101 pw-class mpls
```

```
16:57:34: SSM CM: provision switch event, switch id 86040
16:57:34: SSM CM: [Ethernet] provision first segment, id 12313
16:57:34: SSM CM: CM FSM: state Idle - event Provision segment
16:57:34: SSM CM: [SSS:Ethernet:12313] provision segment 1
16:57:34: SSM SM: [SSS:Ethernet:12313] event Provision segment
16:57:34: SSM CM: [SSS:Ethernet] shQ request send ready event
16:57:34: SSM CM: SM msg event send ready event
16:57:34: SSM SM: [SSS:Ethernet:12313] segment ready
16:57:34: SSM SM: [SSS:Ethernet:12313] event Found segment data
16:57:34: SSM CM: Query AToM to Ethernet switching, enabled
16:57:34: SSM CM: [AToM] provision second segment, id 16410
16:57:34: SSM CM: CM FSM: state Down - event Provision segment
16:57:34: SSM CM: [SSS:AToM:16410] provision segment 2
16:57:34: SSM SM: [SSS:AToM:16410] event Provision segment
16:57:34: SSM CM: [AToM] send client event 6, id 16410
16:57:34: label_oce_get_label_bundle: flags 14 label 19
16:57:34: SSM CM: [SSS:AToM] shQ request send ready event
16:57:34: SSM CM: SM msg event send ready event
16:57:34: SSM SM: [SSS:AToM:16410] segment ready
16:57:34: SSM SM: [SSS:AToM:16410] event Found segment data
16:57:34: SSM SM: [SSS:AToM:16410] event Bind segment
16:57:34: SSM SM: [SSS:Ethernet:12313] event Bind segment
16:57:34: SSM CM: [AToM] send client event 3, id 16410
```

```
Router# configure terminal
```

```
Router(config)# interface e1/0
```

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

```
16:57:26: SSM CM: [Ethernet] unprovision segment, id 16387
16:57:26: SSM CM: CM FSM: state Open - event Free segment
16:57:26: SSM CM: [SSS:Ethernet:16387] unprovision segment 1
16:57:26: SSM SM: [SSS:Ethernet:16387] event Unprovision segment
```

```

16:57:26: SSM CM: [SSS:Ethernet] shQ request send unprovision complete event
16:57:26: SSM CM: [SSS:AToM:86036] unbind segment 2
16:57:26: SSM SM: [SSS:AToM:86036] event Unbind segment
16:57:26: SSM CM: SM msg event send unprovision complete event
16:57:26: SSM SM: [SSS:Ethernet:16387] free segment class
16:57:26: SSM SM: [SSS:Ethernet:16387] free segment
16:57:26: SSM SM: [SSS:Ethernet:16387] event Free segment
16:57:26: SSM SM: last segment class freed
16:57:26: SSM CM: unprovision switch event, switch id 12290
16:57:26: SSM CM: [SSS:AToM] shQ request send unready event
16:57:26: SSM CM: SM msg event send unready event
16:57:26: SSM SM: [SSS:AToM:86036] event Unbind segment
16:57:26: SSM CM: [AToM] unprovision segment, id 86036
16:57:26: SSM CM: CM FSM: state Down - event Free segment
16:57:26: SSM CM: [SSS:AToM:86036] unprovision segment 2
16:57:26: SSM SM: [SSS:AToM:86036] event Unprovison segment
16:57:26: SSM CM: [SSS:AToM] shQ request send unprovision complete event
16:57:26: SSM CM: SM msg event send unprovision complete event
16:57:26: SSM SM: [SSS:AToM:86036] free segment class
16:57:26: SSM SM: [SSS:AToM:86036] free segment
16:57:26: SSM SM: [SSS:AToM:86036] event Free segment
16:57:26: SSM SM: last segment class freed

```

Related Commands

Command	Description
show ssm	Displays SSM information for switched Layer 2 segments.

debug sss aaa authorization event



Note

Effective with Cisco IOS Release 15.0(1)S, the **debug sss aaa authorization event** command is replaced by the **debug subscriber aaa authorization event** command. See the **debug subscriber aaa authorization event command** for more information.

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.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)S	This command was replaced by the debug subscriber aaa authorization event command.

Examples

The following is sample output of several Subscriber Service Switch (SSS) **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
nobody@example.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 'example.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 example.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 example.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



Note

Effective with Cisco IOS Release 15.0(1)S, the **debug sss aaa authorization fsm** command is replaced by the **debug subscriber aaa authorization fsm** command. See the **debug subscriber aaa authorization fsm** command for more information.

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.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)S	This command was replaced by the debug subscriber aaa authorization fsm command.

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.

Command	Description
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



Note

Effective with Cisco IOS Release 15.0(1)S, the **debug sss error** command is replaced by the **debug subscriber error** command. See the **debug subscriber error** command for more information.

To display diagnostic information about errors that may occur during Subscriber Service Switch (SSS) 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.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)S	This command was replaced by the debug subscriber error command.

Examples

The following example shows how to enter this command. See the “Examples” section of the [debug ssg transparent login](#) 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.

Command	Description
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



Note

Effective with Cisco IOS Release 15.0(1)S, the **debug sss event** command is replaced by the **debug subscriber event** command. See the **debug subscriber event** command for more information.

To display diagnostic information about Subscriber Service Switch (SSS) 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.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)S	This command was replaced by the debug subscriber event command.

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.

Command	Description
<code>debug sss error</code>	Displays diagnostic information about errors that may occur during Subscriber Service Switch call setup.
<code>debug sss fsm</code>	Displays diagnostic information about the Subscriber Service Switch call setup state.

debug sss fsm



Note

Effective with Cisco IOS Release 15.0(1)S, the **debug sss fsm** command is replaced by the **debug subscriber fsm** command. See the **debug subscriber fsm** command for more information.

To display diagnostic information about the Subscriber Service Switch (SSS) 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.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)S	This command was replaced by the debug subscriber fsm command.

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
```

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.

Command	Description
<code>debug sss error</code>	Displays diagnostic information about errors that may occur during Subscriber Service Switch call setup.
<code>debug sss event</code>	Displays diagnostic information about the Subscriber Service Switch call setup events.

debug standby

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

debug standby [terse]

no debug standby [terse]

Syntax Description

terse (Optional) Displays a limited range of HSRP errors, events, and packets.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

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 329 describes the significant fields shown in the display.

Table 329 *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 <i>address</i>	Hot Standby group IP address advertised in the hello packet.
state	Transition from one state to another.
Coup out <i>address</i>	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
```

Related Commands

Command	Description
debug condition standby	Filters the output of the debug standby command on the basis of HSRP group number.
debug standby errors	Displays error messages related to HSRP.
debug standby events	Displays events related to HSRP.
debug standby events icmp	Displays debugging messages for the HSRP ICMP redirects filter.
debug standby packets	Displays debugging information for packets related to HSRP.

debug standby errors

To display error messages related to Host Standby Router Protocol (HSRP), use the **debug standby errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug standby errors

no debug standby errors

Syntax Description This command has no arguments or keywords.

Defaults Debugging is not enabled.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.1	This command was introduced.
	12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
	12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines You can filter the **debug** output using interface and HSRP group conditional debugging. To enable interface conditional debugging, use the **debug condition interface** command. To enable HSRP conditional debugging, use the **debug condition standby** command.

Examples The following example enables the display of HSRP errors:

```
Router# debug standby errors

HSRP Errors debugging is on.
```

Related Commands	Command	Description
	debug condition standby	Filters the output of the debug standby command on the basis of HSRP group number.
	debug standby	Displays HSRP state changes.
	debug standby events	Displays events related to HSRP.
	debug standby events icmp	Displays debugging messages for the HSRP ICMP redirects filter.
	debug standby packets	Displays debugging information for packets related to HSRP.

debug standby events

To display events related to Hot Standby Router Protocol (HSRP), use the **debug standby events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug standby events [all | api | arp | ha | internal {data | init | state | timer} | protocol |
redundancy | terse | track] [detail]
```

```
no debug standby events [all | arp | ha | internal {api | data | init | state | timer} | protocol |
redundancy | terse | track] [detail]
```

Syntax Description

all	(Optional) Displays all HSRP events.
api	(Optional) Displays HSRP application programming interface (API) events.
arp	(Optional) Displays HSRP Address Resolution Protocol (ARP) events.
ha	(Optional) Displays High availability (HA) events.
internal	(Optional) Displays Internal HSRP events.
data	(Optional) Displays HSRP data events.
init	(Optional) Displays HSRP startup and shutdown events.
state	(Optional) Displays HSRP state events.
timer	(Optional) Displays HSRP timer events.
protocol	(Optional) Displays HSRP protocol events.
redundancy	(Optional) Displays HSRP redundancy events.
terse	(Optional) Displays all HSRP packets, except hellos and advertisements.
track	(Optional) Displays HSRP tracking events.
detail	(Optional) Displays detailed debugging information.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.1	This command was introduced.
12.2(8)T	The api keyword was added.
12.4(4)T	The ha keyword was added.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)SXI	The arp keyword was added.
12.4(24)T	This command was modified. The init keyword was added.
12.2(33)SX11	This command was modified. The init keyword was added.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.

Usage Guidelines

You can filter the debug output using interface and HSRP group conditional debugging. To enable interface conditional debugging, use the **debug condition interface** command. To enable HSRP conditional debugging, use the **debug condition standby** command.

Examples

The following example shows how to enable the debugging of the active and standby Route Processors (RPs) on an active RP console. The HSRP group is configured on the active RP, and the HSRP state is active.

```
Router# debug standby events ha

!Active RP
*Apr 27 04:13:47.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Listen into sync buffer
*Apr 27 04:13:47.855: HSRP: CF Sync send ok
*Apr 27 04:13:57.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Speak into sync buffer
*Apr 27 04:13:57.855: HSRP: CF Sync send ok
*Apr 27 04:14:07.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Standby into sync buffer
*Apr 27 04:14:07.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Active into sync buffer
*Apr 27 04:14:07.863: HSRP: CF Sync send ok
*Apr 27 04:14:07.867: HSRP: CF Sync send ok

!Standby RP
*Apr 27 04:11:21.011: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:21.011: HSRP: Gi0/0/1 Grp 101 RF sync state Init -> Listen
*Apr 27 04:11:31.011: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:31.011: HSRP: Gi0/0/1 Grp 101 RF sync state Listen -> Speak
*Apr 27 04:11:41.071: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:41.071: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:41.071: HSRP: Gi0/0/1 Grp 101 RF sync state Speak -> Standby
*Apr 27 04:11:41.071: HSRP: Gi0/0/1 Grp 101 RF sync state Standby -> Active
```

[Table 330](#) describes the significant fields shown in the display.

Table 330 debug standby events Field Descriptions

Field	Description
RF	Redundancy facility—Internal mechanism that makes Stateful Switchover (SSO) work.
CF	Checkpoint facility—Internal mechanism that makes SSO work.

The following sample shows HSRP debug information when HSRP is configured to send gratuitous ARP packets every four seconds:

```
Router# debug standby event arp detail

HSRP Events debugging is on (arp)

*Jun 27 14:15:51.795: HSRP: Et0/0 Grp 1 Send grat ARP 10.0.0.1 mac 0000.0c07.ac01 (use vMAC)
*Jun 27 14:15:55.755: HSRP: Et0/0 Grp 1 Send grat ARP 10.0.0.1 mac 0000.0c07.ac01 (use vMAC)
*Jun 27 14:15:59.407: HSRP: Et0/0 Grp 1 Send grat ARP 10.0.0.1 mac 0000.0c07.ac01 (use vMAC)
```

**Note**

Debug messages for gratuitous ARP packets are seen only if the **detail** keyword is entered.

Table 331 describes the significant fields shown in the display.

Table 331 *debug standby events detail Field Descriptions*

Field	Description
Send grat ARP 10.0.0.1	IP address to which HSRP sends gratuitous ARP packets.
mac	MAC address of the host router to which HSRP sends gratuitous ARP packets.

The following examples show the output of the **debug standby event internal init** command when the IP address of an interface is changed and HSRP makes an internal evaluation to see if the added address permits the currently configured standby address to remain valid.

Router# **debug standby events internal init**

```
HSRP: Ethernet0/0 vIP intf primary subnet 172.24.1.0 added
.
.
.
HSRP: Ethernet0/0 vIP 172.24.1.254 matches intf primary subnet 172.24.1.0
```

Router# **debug standby events internal init**

```
HSRP: Ethernet0/0 vIP intf secondary subnet 172.24.1.0 added
.
.
.
HSRP: Ethernet0/0 vIP 172.24.1.254 matches intf secondary subnet 172.24.1.0
```

Router# **debug standby events internal init**

```
HSRP: Ethernet0/0 vIP intf secondary subnet 172.24.1.0 deleted
.
.
.
HSRP: Ethernet0/0 vIP 172.24.1.254 matches no intf subnets
```

Related Commands

Command	Description
debug condition interface	Limits output for some debug commands on the basis of the interface, VC, or VLAN.
debug condition standby	Filters the output of the debug standby command on the basis of HSRP group number.
debug standby	Displays HSRP state changes.
debug standby errors	Displays error messages related to HSRP.
debug standby events icmp	Displays debugging messages for the HSRP ICMP redirects filter.
debug standby packets	Displays debugging information for packets related to HSRP.

debug standby events icmp

To display debugging 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.
	12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
	Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.

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 debugging 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 standby events neighbor

To display Hot Standby Router Protocol (HSRP) Bidirectional Forwarding Detection (BFD) peering events, use the **debug standby events neighbor** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug standby events neighbor

no debug standby events neighbor

Syntax Description This command has no arguments or keywords.

Command Default HSRP neighbor debugging output is not displayed.

Command Modes Privileged EXEC (#)

Command History	Release	Modification
	12.4(11)T	This command was introduced.
	12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines You can filter the debug output using interface and HSRP group conditional debugging. To enable interface conditional debugging, use the **debug condition** interface command. To enable HSRP conditional debugging, use the **debug condition standby** command.

Examples In this example, two HSRP routers are configured as neighbors, supporting BFD peering with the **debug standby events neighbor** command configured. The following example shows the debug output that appears when an additional HSRP group is added to Router A:

Router A

```
RouterA# debug standby event neighbor
```

```
HSRP Events debugging is on
(neighbor)
*Oct  3 02:57:48.587: HSRP: Fa2/0 Grp 2 Standby router is local
01:03:49: %HSRP-5-STATECHANGE: FastEthernet2/0 Grp 2 state Speak -> Standby
*Oct  3 02:57:49.087: HSRP: Fa2/0 Grp 2 Active router is local
*Oct  3 02:57:49.087: HSRP: Fa2/0 Grp 2 Standby router is unknown, was local
01:03:50: %HSRP-5-STATECHANGE: FastEthernet2/0 Grp 2 state Standby -> Active
```

Router B

```
RouterB# debug standby event neighbor
```

```

HSRP Events debugging is on
(neighbor)
*Oct 3 10:00:28.503: HSRP: Fa2/0 Grp 2 Active router is 10.0.0.1 (no local config)
*Oct 3 10:00:28.503: HSRP: Fa2/0 Nbr 10.0.0.1 active for group 2

```

The following example shows the debug output when an additional HSRP group is added to Router B:

Router B

```

*Oct 3 10:02:28.067: HSRP: Fa2/0 Nbr 10.0.0.1 no longer active for group 2 (Disabled)
*Oct 3 10:02:28.503: HSRP: Fa2/0 Grp 2 Active router is 10.0.0.1
*Oct 3 10:02:28.503: HSRP: Fa2/0 Nbr 10.0.0.1 active for group 2
*Oct 3 10:02:48.071: HSRP: Fa2/0 Grp 2 Standby router is local
00:44:28: %HSRP-5-STATECHANGE: FastEthernet2/0 Grp 2 state Speak -> Standby

```

Router A

```

*Oct 3 03:00:08.655: HSRP: Fa2/0 Grp 2 Standby router is 10.0.0.2
*Oct 3 03:00:08.655: HSRP: Fa2/0 Nbr 10.0.0.2 standby for group 2

```

The following is sample debug output showing a possible network outage (the loss of signal between the ports of Router A and B):

Router B

```

*Oct 3 10:09:07.651: HSRP: Fa2/0 Grp 1 Active router is local, was 10.0.0.1
*Oct 3 10:09:07.651: HSRP: Fa2/0 Nbr 10.0.0.1 no longer active for group 1 (Standby)
*Oct 3 10:09:07.651: HSRP: Fa2/0 Grp 1 Standby router is unknown, was local
00:50:48: %HSRP-5-STATECHANGE: FastEthernet2/0 Grp 1 state Standby -> Active
*Oct 3 10:09:08.959: HSRP: Fa2/0 Grp 2 Active router is local, was 10.0.0.1
*Oct 3 10:09:08.959: HSRP: Fa2/0 Nbr 10.0.0.1 no longer active for group 2 (Standby)
*Oct 3 10:09:08.959: HSRP: Fa2/0 Nbr 10.0.0.1 Was active or standby - start passive
holddown
*Oct 3 10:09:08.959: HSRP: Fa2/0 Grp 2 Standby router is unknown, was local
00:50:49: %HSRP-5-STATECHANGE: FastEthernet2/0 Grp 2 state Standby -> Active

```

Related Commands

Command	Description
debug bfd	Displays debugging messages about BFD.
debug condition	Limits the output for some debug commands based on specified conditions.
debug condition standby	Limits the debugging output of HSRP state changes.
show bfd neighbor	Displays a line-by-line listing of existing BFD adjacencies.
show standby	Displays HSRP information.
show standby neighbors	Displays information about HSRP neighbors.
standby bfd all-interfaces	Reenables HSRP BFD peering on all interfaces if it has been disabled.
standby ip	Activates HSRP.

debug standby packets

To display debugging information for packets related to Hot Standby Router Protocol (HSRP), use the **debug standby packets** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug standby packets [advertise | all | terse | coup | hello | resign] [detail]

no debug standby packet [advertise | all | terse | coup | hello | resign] [detail]

Syntax Description

advertise	(Optional) Specifies HSRP advertisement packets.
all	(Optional) Specifies all HSRP packets.
terse	(Optional) Specifies all HSRP packets, except hellos and advertisements.
coup	(Optional) Specifies HSRP coup packets.
hello	(Optional) Specifies HSRP hello packets.
resign	(Optional) Specifies HSRP resign packets.
detail	(Optional) Specifies HSRP packets in detail.

Defaults

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1	This command was introduced.
12.2	The advertise keyword was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

You can filter the debug output using interface and HSRP group conditional debugging. To enable interface conditional debugging, use the **debug condition interface** command. To enable HSRP conditional debugging, use the **debug condition standby** command.



Note

HSRP advertisement packets are packets that are related to HSRP interfaces. Other packet types, including, hello, coup, and resign packets relate to an HSRP group.

Examples

The following example show how to enable the display of all HSRP packets:

```
Router# debug standby packets all
```

```
HSRP Packets debugging is on.
```

Related Commands

Command	Description
debug condition interface	Limits output for some debugging commands based on the interfaces.
debug condition standby	Filters the output of the debug standby command on the basis of HSRP group number.
debug standby	Displays HSRP state changes.
debug standby errors	Displays error messages related to HSRP.
debug standby events	Displays events related to HSRP.
debug standby events icmp	Displays debugging messages for the HSRP ICMP redirects filter.

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

<i>group</i>	(Optional) A decimal integer assigned to a group. Using this option limits output to packets associated with the specified STUN group.
<i>address</i>	(Optional) The output is further limited to only those packets containing the specified STUN address. The <i>address</i> 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
X1 type of 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
X2 type of packet — 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
X3 type of packet — 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

```

S2563

The following line describes an X1 type of packet:

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

Table 332 describes the significant fields in this line of **debug stun packet** output.

Table 332 *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 Synchronous Data Link Control (SDL) end nodes. Possible values are as follows: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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 subscriber aaa authorization

To display diagnostic information about authentication, authorization, and accounting (AAA) authorization of Intelligent Service Gateway (ISG) subscriber sessions, use the **debug subscriber aaa authorization** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug subscriber aaa authorization { event | fsm }
```

```
no debug sss aaa authorization { event | fsm }
```

Syntax Description

event	Display information about AAA authorization events that occur during ISG session establishment.
fsm	Display information about AAA authorization state changes for ISG subscriber sessions.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(28)SB	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

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

```
Router# debug subscriber event
Router# debug subscriber error
Router# debug subscriber state
Router# debug subscriber aaa authorization event
Router# debug subscriber 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 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 subscriber error

To display diagnostic information about errors that may occur during Intelligent Service Gateway (ISG) subscriber session setup, use the **debug subscriber error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug subscriber error

no debug subscriber 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(28)SB	This command was introduced.
	12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples The following sample output for the **debug subscriber error** command indicates that the session is stale since the session handle has already been destroyed.

```
Router# debug subscriber error
```

```
*Sep 20 22:39:49.455: SSS MGR: Session handle [EF000002] destroyed already
```

Related Commands	Command	Description
	debug sss aaa authorization event	Displays messages about AAA authorization events that are part of normal call establishment.
	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 subscriber event

To display diagnostic information about Intelligent Service Gateway (ISG) subscriber session setup events, use the **debug subscriber event** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug subscriber event

no debug subscriber 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(28)SB	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

The following sample output for the **debug subscriber event** commands indicates that the system has determined that the session should be locally terminated. The local termination module determines that an interface description block (IDB) is not required for this session, and it sets up the data plane for packet switching.

```
Router# debug subscriber event
```

```
*Sep 20 22:21:08.223: SSS MGR [uid:2]: Handling Connect Local Service action
*Sep 20 22:21:08.223: SSS LTERM [uid:2]: Processing Local termination request
*Sep 20 22:21:08.223: SSS LTERM [uid:2]: L3 session - IDB not required for setting up
service
*Sep 20 22:21:08.223: SSS LTERM [uid:2]: Interface already present or not required for
service
*Sep 20 22:21:08.223: SSS LTERM [uid:2]: Segment provision successful
```

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 fsm	Displays diagnostic information about the Subscriber Service Switch call setup state.

debug subscriber feature

To display diagnostic information about the installation and removal of Intelligent Service Gateway (ISG) features on Intelligent Service Gateway (ISG) subscriber sessions, use the **debug subscriber feature** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug subscriber feature { **all** | **detail** | **error** | **event** | **name** *name-of-feature* { **detail** | **error** | **event** | **packet** } | **packet** [**detail** | **full**]

no debug subscriber feature { **all** | **detail** | **error** | **event** | **name** *name-of-feature* { **detail** | **error** | **event** | **packet** } | **packet** [**detail** | **full**]

Syntax Description	
all	Displays information about all features.
detail	<p>The detail keyword can be used in one of the following three ways:</p> <ul style="list-style-type: none"> • If used with no other keywords, displays detailed information about all features • If a feature name is specified with the name <i>name-of-feature</i> keyword and argument, displays detailed information about the specific feature. The detail keyword can be used with the following <i>name-of-feature</i> values: <ul style="list-style-type: none"> – accounting – compression – modem-on-hold – policing – traffic-classification • If used with the packet keyword, displays a partial dump of packets as ISG features are being applied to the packets.
error	Displays information about errors for all features or a specified feature.
event	Displays information about events for all features or a specified feature.
name	Displays information about a specific feature.

<i>name-of-feature</i>	Name of the ISG feature. Possible values are the following: <ul style="list-style-type: none"> • access-list • accounting • compression • filter • idle-timer • interface-config • ip-config • l4redirect • modem-on-hold • policing • portbundle • prepaid-idle • session-timer • static-routes • time-monitor • traffic-classification • volume-monitor
packet	Displays information about packets as ISG features are being applied to the packets. If a feature name is specified with the name <i>name-of-feature</i> keyword and argument, packet information about the specific feature is displayed. The packet keyword can be used with the following <i>name-of-feature</i> values: <ul style="list-style-type: none"> • access-list • l4redirect • policing • portbundle
full	(Optional) Displays a full dump of a packet as ISG features are being applied to it.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(28)SB	This command was introduced.
	12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

The following sample output for the **debug subscriber feature** command indicates that the idle timeout feature has been successfully installed on the inbound segment.

```
Router# debug subscriber feature event
```

```
*Sep 20 22:28:57.903: SSF[myservice/uid:6/Idle Timeout]: Group feature install
```

```
*Sep 20 22:28:57.903: SSF[uid:6/Idle Timeout]: Adding feature to inbound segment(s)
```

debug subscriber fsm

To display diagnostic information about Intelligent Service Gateway (ISG) subscriber session state change, use the **debug subscriber fsm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug subscriber fsm

no debug subscriber 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(28)SB	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

The following sample output for the **debug subscriber fsm** command indicates that the session has been disconnected by the client, and the system is cleaning up the session by disconnecting the network service and removing any installed features.

```
Router# debug subscriber fsm
```

```
*Sep 20 22:35:10.495: SSS MGR [uid:5]: Event client-disconnect, state changed from  
connected to disconnecting-fsp-feat
```

debug subscriber packet

To display information about packets as they traverse the subscriber service switch (SSS) path, use the **debug subscriber packet** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug subscriber packet { **detail** | **error** | **event** | **full** }

no debug subscriber packet { **detail** | **error** | **event** | **full** }

Syntax Description

detail	Displays a partial dump of packets as they traverse the SSS path.
error	Displays any packet-switching errors that occur when a packet traverses the SSS path.
event	Displays packet-switching events that occur when a packet traverses the SSS path.
full	Displays a full dump of packets as they traverse the SSS path.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(28)SB	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

The following example show sample output for the **debug subscriber packet** command with the **full** keyword. This output is for a PPPoE session configured with forwarding.

```
SSS Switch: Pak encap size, old: 60, new: 24
SSS Switch: Pak 0285C458 sz 66 encap 14
*Feb  9 15:47:13.659: 000000 AA BB CC 00 0B 01 AA BB D.....
*Feb  9 15:47:13.659: 000008 CC 00 0C 01 08 00 45 00 .....N.
*Feb  9 15:47:13.659: 000010 00 34 00 28 00 00 FE 11 .4.(....
*Feb  9 15:47:13.659: 000018 F2 9D AC 12 B8 E7 AC 12 .....
*Feb  9 15:47:13.659: 000020 B8 E6 06 A5 06 A5 00 20 .....
*Feb  9 15:47:13.659: 000028 00 00 C0 01 02 00 00 02 .....
*Feb  9 15:47:13.659: 000030 00 01 00 18 00 00 FC A7 .....
*Feb  9 15:47:13.659: 000038 2E B3 FF 03 C2 23 03 01 .....#.
*Feb  9 15:47:13.659: 000040 00 04 ..
SSS Switch: Pak encap size, old: 60, new: 24
SSS Switch: Pak 0285C458 sz 72 encap 14
*Feb  9 15:47:13.691: 000000 AA BB CC 00 0B 01 AA BB D.....
*Feb  9 15:47:13.691: 000008 CC 00 0C 01 08 00 45 00 .....N.
*Feb  9 15:47:13.691: 000010 00 3A 00 2A 00 00 FE 11 ..*....
*Feb  9 15:47:13.691: 000018 F2 95 AC 12 B8 E7 AC 12 .....
*Feb  9 15:47:13.691: 000020 B8 E6 06 A5 06 A5 00 26 .....&
*Feb  9 15:47:13.691: 000028 00 00 C0 01 02 00 00 02 .....
*Feb  9 15:47:13.691: 000030 00 01 00 1E 00 00 FC A7 .....
*Feb  9 15:47:13.691: 000038 2E B3 FF 03 80 21 01 01 .....!..
```

```

*Feb  9 15:47:13.691: 000040 00 0A 03 06 3A 3A 3A 3A  ....:
SSS Switch: Pak encap size, old: 24, new: 46
SSS Switch: Pak 027A5BE8 sz 36 encap 18
*Feb  9 15:47:13.691: 000000 AA BB CC 00 0B 00 AA BB  D.....
*Feb  9 15:47:13.691: 000008 CC 00 0A 00 81 00 01 41  .....a
*Feb  9 15:47:13.691: 000010 88 64 11 00 00 01 00 0C  .dN.....
*Feb  9 15:47:13.691: 000018 80 21 01 01 00 0A 03 06  .!.....
*Feb  9 15:47:13.691: 000020 00 00 00 00  ....
SSS Switch: Pak encap size, old: 60, new: 24
SSS Switch: Pak 0285C458 sz 72 encap 14
*Feb  9 15:47:13.691: 000000 AA BB CC 00 0B 01 AA BB  D.....
*Feb  9 15:47:13.691: 000008 CC 00 0C 01 08 00 45 00  .....N.
*Feb  9 15:47:13.691: 000010 00 3A 00 2C 00 00 FE 11  .:.,....
*Feb  9 15:47:13.691: 000018 F2 93 AC 12 B8 E7 AC 12  .....
*Feb  9 15:47:13.691: 000020 B8 E6 06 A5 06 A5 00 26  .....&
*Feb  9 15:47:13.691: 000028 00 00 C0 01 02 00 00 02  .....
*Feb  9 15:47:13.691: 000030 00 01 00 1E 00 00 FC A7  .....
*Feb  9 15:47:13.691: 000038 2E B3 FF 03 80 21 03 01  .....!..
*Feb  9 15:47:13.691: 000040 00 0A 03 06 09 00 00 1F  .....

```

Related Commands

Command	Description
debug subscriber feature	Displays diagnostic information about the installation and removal of ISG features on subscriber sessions.

debug subscriber policy

To display diagnostic information about policy execution related to Intelligent Service Gateway (ISG) subscriber sessions, use the **debug subscriber policy** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug subscriber policy {all | detail | error | event | fsm | prepaid | {condition | idmgr | profile
| push | rule | service} [detail | error | event] | dpm [error | event] | webportal {detail | error
| event}}
```

```
no debug subscriber policy {all | detail | error | event | fsm | prepaid | {condition | idmgr | profile
| push | rule | service} [detail | error | event] | dpm [error | event] | webportal {detail | error
| event}}
```

Syntax Description

all	Displays information about all policies.
detail	Displays detailed information about all policies or the specified type of policy.
error	Displays policy execution errors for all policies or the specified type of policy.
event	Displays policy execution events for all policies or the specified type of policy.
fsm	Displays information about state changes during policy execution.
prepaid	Displays information about ISG prepaid policy execution.
condition	Displays information related to the evaluation of ISG control class maps.
idmgr	Displays information about policy execution related to identity.
profile	Displays information about the policy manager subscriber profile database.
push	Displays policy information about dynamic updates to subscriber profiles from policy servers.
rule	Displays information about control policy rules.
service	Displays policy information about service profile database events for subscriber sessions.
dpm	Displays information about Dynamic Host Configuration Protocol (DHCP) in relation to subscriber sessions.
webportal	Displays policy information about the web portal in relation to subscriber sessions.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(28)SB	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

The following example shows sample output for the **debug subscriber policy** command with the **events** keyword. This output indicates the creation of a new session. "Updated key list" indicates important attributes and information associated with the session.

```
*Feb 7 18:58:24.519: SSS PM [0413FC58]: Create context 0413FC58
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Authen status update; is now "unauthen"
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Updated NAS port for AAA ID 14
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Updated key list:
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Access-Type = 15 (IP)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Protocol-Type = 4 (IP)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Media-Type = 2 (IP)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   IP-Address = 10.0.0.2 (0A000002)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   IP-Address-VRF = IP 10.0.0.2:0
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   source-ip-address = 037FBB78
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Mac-Address = aabb.cc00.6500
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Final = 1 (YES)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Authen-Status = 1 (Unauthenticated)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Nasport = PPPoEoE: slot 0 adapter 0 port
0
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Updated key list:
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Access-Type = 15 (IP)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Protocol-Type = 4 (IP)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Media-Type = 2 (IP)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   IP-Address = 10.0.0.2 (0A000002)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   IP-Address-VRF = IP 10.0.0.2:0
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   source-ip-address = 037FBB78
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Mac-Address = aabb.cc00.6500
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Final = 1 (YES)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Authen-Status = 1 (Unauthenticated)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Nasport = PPPoEoE: slot 0 adapter 0 port
0
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]:   Session-Handle = 486539268 (1D000004)
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: SM Policy invoke - Service Selection
Request
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Access type IP
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Access type IP: final key
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Received Service Request
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Handling Authorization Check
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: SIP [IP] can NOT provide more keys
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: SIP [IP] can NOT provide more keys
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Handling Default Service
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Providing Service
*Feb 7 18:58:24.519: SSS PM [uid:4][0413FC58]: Policy reply - Local Terminate
*Feb 7 18:58:24.523: SSS PM [uid:4][0413FC58]: SM Policy invoke - Apply Config Success
*Feb 7 18:58:24.523: SSS PM [uid:4][0413FC58]: Handling Apply Config; SUCCESS
```

debug subscriber service

To display diagnostic information about the service profile database in an Intelligent Service Gateway (ISG), use the **debug subscriber service** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug subscriber service

no debug subscriber service

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(28)SB	This command was introduced.
	12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines Use the **debug subscriber service** command to diagnose problems with service profiles or service policy maps.

Examples The following example shows sample output for the **debug subscriber service** command. This output indicates that a service logon has occurred for the service “prep_service”.

```
*Feb 7 18:52:31.067: SVM [prep_service]: needs downloading
*Feb 7 18:52:31.067: SVM [D6000000/prep_service]: allocated version 1
*Feb 7 18:52:31.067: SVM [D6000000/prep_service]: [8A000002]: client queued
*Feb 7 18:52:31.067: SVM [D6000000/prep_service]: [PM-Download:8A000002] locked 0->1
*Feb 7 18:52:31.067: SVM [D6000000/prep_service]: [AAA-Download:040DD9D0] locked 0->1
*Feb 7 18:52:31.127: SVM [D6000000/prep_service]: TC feature info found
*Feb 7 18:52:31.127: SVM [D0000001/prep_service]: added child
*Feb 7 18:52:31.127: SVM [D6000000/prep_service]: [TC-Child:040DD130] locked 0->1
*Feb 7 18:52:31.127: SVM [D0000001/CHILD/prep_service]: [TC-Parent:040DD1A8] locked 0->1
*Feb 7 18:52:31.127: SVM [D6000000/prep_service]: TC flow feature info not found
*Feb 7 18:52:31.127: SVM [D6000000/prep_service]: downloaded first version
*Feb 7 18:52:31.127: SVM [D6000000/prep_service]: [8A000002]: client download ok
*Feb 7 18:52:31.127: SVM [D6000000/prep_service]: [SVM-to-client-msg:8A000002] locked 0->1
*Feb 7 18:52:31.127: SVM [D6000000/prep_service]: [AAA-Download:040DD9D0] unlocked 1->0
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: alloc feature info
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: [SVM-Feature-Info:040E2E80] locked 0->1
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: has Policy info
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: [PM-Info:0416BAB0] locked 0->1
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: populated client
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: [PM-Download:8A000002] unlocked 1->0
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: [SVM-to-client-msg:8A000002] unlocked
1->0
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: [PM-Service:040E31E0] locked 0->1
```

```
*Feb 7 18:52:31.131: SVM [D0000001/CHILD/prep_service]: [SM-SIP-Apply:D0000001] locked
0->1
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: [FM-Bind:82000002] locked 0->1
*Feb 7 18:52:31.131: SVM [D6000000/prep_service]: [SVM-Feature-Info:040E2E80] unlocked
1->0
*Feb 7 18:52:31.139: SVM [D0000001/CHILD/prep_service]: alloc feature info
*Feb 7 18:52:31.139: SVM [D0000001/CHILD/prep_service]: [SVM-Feature-Info:040E2E80] locked
0->1
*Feb 7 18:52:31.159: SVM [D0000001/CHILD/prep_service]: [FM-Bind:2C000003] locked 0->1
*Feb 7 18:52:31.159: SVM [D0000001/CHILD/prep_service]: [SVM-Feature-Info:040E2E80]
unlocked 1->0
*Feb 7 18:52:31.159: SVM [D0000001/CHILD/prep_service]: [SM-SIP-Apply:D0000001] unlocked
1->0
```

debug subscriber testing

To display diagnostic information for Intelligent Service Gateway (ISG) simulator testing, use the **debug subscriber testing** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug subscriber testing

no debug subscriber testing

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(28)SB	This command was introduced.
	12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples The following example shows the configuration of the **debug subscriber testing** command:

```
Router# debug subscriber testing
```

debug sw56

To display debugging information for switched 56K services, use the **debug sw56** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug sw56

no debug sw56

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

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 msec = 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 is sample output from the **debug syslog-server** command. The sample output shows 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 333](#) describes the significant fields shown in the display.

Table 333 *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 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 debugging information for switched 56K services.

debug tag-switching atm-cos

The **debug tag-switching atm-cos** command is replaced by the **debug mpls atm-cos** command. See the **debug mpls atm-cos** command for more information.

debug tag-switching atm-tdp api

The **debug tag-switching atm-tdp api** command is replaced by the **debug mpls atm-ldp api** command. See the **debug mpls atm-ldp api** command for more information.

debug tag-switching atm-tdp routes

The **debug tag-switching atm-tdp routes** command is replaced by the **debug mpls atm-ldp routes** command. See the **debug mpls atm-ldp routes** command for more information.

debug tag-switching atm-tdp states

The **debug tag-switching atm-tdp states** command is replaced by the **debug mpls atm-ldp states** command. See the **debug mpls atm-ldp states** command for more information.

debug tag-switching tdp advertisements

The **debug tag-switching tdp advertisements** command is replaced by the **debug mpls ldp advertisements** command. See the **debug mpls ldp advertisements** command for more information.

debug tag-switching tdp bindings

The **debug tag-switching tdp bindings** command is replaced by the **debug mpls ldp bindings** command. See the **debug mpls ldp bindings** command for more information.

debug tag-switching tdp directed-neighbors

The **debug tag-switching tdp directed-neighbors** command is replaced by the **debug mpls ldp targeted-neighbors** command. See the **debug mpls ldp targeted-neighbors** command for more information.

debug tag-switching tdp peer state-machine

The **debug tag-switching tdp peer state-machine** command is replaced by the **debug mpls ldp peer state-machine** command. See the **debug mpls ldp peer state-machine** command for more information.

debug tag-switching tdp pies received

The **debug tag-switching tdp pies received** command is replaced by the **debug mpls ldp session io** command. See the **debug mpls ldp session io** command for more information.

debug tag-switching tdp pies sent

The **debug tag-switching tdp pies sent** command is replaced by the **debug mpls ldp messages** command. See the **debug mpls ldp messages** command for more information.

debug tag-switching tdp session io

The **debug tag-switching tdp session io** command is replaced by the **debug mpls ldp session io** command. See the **debug mpls ldp session io** command for more information

debug tag-switching tdp session state-machine

The **debug tag-switching tdp session state-machine** command is replaced by the **debug mpls ldp session state-machine** command. See the **debug mpls ldp session state-machine** command for more information.

debug tag-switching tdp transport connections

The **debug tag-switching tdp transport connections** command is replaced by the **debug mpls ldp transport connections** command. See the **debug mpls ldp transport connections** command for more information.

debug tag-switching tdp transport events

The **debug tag-switching tdp transport events** command is replaced by the **debug mpls ldp transport events** command. See the **debug mpls ldp transport events** command for more information.

debug tag-switching tdp transport timers

To print information about events that restart the “hold” timers that are part of the TDP discovery mechanism, use the **debug tag-switching tdp transport timers** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug tag-switching tdp transport timers

no debug tag-switching tdp transport timers

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

TDP sessions are supported by data structures and state machines at three levels:

- Transport—The transport level establishes and maintains TCP connections used to support TDP sessions.
- Protocol—The protocol level implements the TDP session setup protocol. The construction and parsing of TDP PDUs and PIEs occur at this level.
- Tag distribution—The tag distribution level uses TDP sessions to exchange tags with TDP peer devices.

The **debug tag-switching tdp transport** command provides visibility of activity at the transport level, the **debug tag-switching tdp session** command at the protocol level, and the **debug tag-switching tdp peer** command at the tag distribution level.

Examples

The following is sample output from the **debug tag-switching tdp transport timers** command:

```
Router# debug tag-switching tdp transport timers

tdp: Start holding timer; adj 0x60D5BC10, 200.26.0.4
tdp: Start holding timer; adj 0x60EA9360, 10.105.0.9
tdp: Start holding timer; adj 0x60D5BC10, 200.26.0.4
tdp: Start holding timer; adj 0x60EA9360, 10.105.0.9
tdp: Start holding timer; adj 0x60D5BC10, 200.26.0.4
tdp: Start holding timer; adj 0x60EA9360, 10.105.0.9
```

[Table 334](#) describes the significant fields shown in the display.

Table 334 *debug tag-switching tdp transport timers Field Descriptions*

Field	Description
tdp	Identifies the source of the message as TDP.
adj 0xnmmmmmm	Identifies the data structure used to represent the peer device at the transport level.
a.b.c.d	Network address of the peer device.

Related Commands	Command	Description
	debug tag-switching tdp transport events	Prints information about the events related to the TDP peer discovery mechanism, which is used to determine the devices with which to establish TDP sessions.

debug tag-switching xtagatm cross-connect

The **debug tag-switching xtagatm cross-connect** command is replaced by the **debug mpls xtagatm cross-connect** command. See the **debug mpls xtagatm cross-connect** command for more information.

debug tag-switching xtagatm errors

The **debug tag-switching xtagatm errors** command is replaced by the **debug mpls xtagatm errors** command. See the **debug mpls xtagatm errors** command for more information.

debug tag-switching xtagatm events

The **debug tag-switching xtagatm events** command is replaced by the **debug mpls xtagatm events** command. See the **debug mpls xtagatm events** command for more information.

debug tag-switching xtagatm vc

The **debug tag-switching xtagatm vc** command is replaced by the **debug mpls xtagatm vc** command. See the **debug mpls xtagatm vc** command for more information.