## Contents

MAC authentication debugging commands
  - debugging mac-authentication .................................................. 1

MAC-in-MAC debugging commands .................................................. 6
  - debugging minm ................................................................. 6

MBGP debugging commands
  - debugging bgp update ipv4 multicast ....................................... 13

MFF debugging commands
  - debugging mac-forced-forwarding ......................................... 16

MLD debugging commands
  - debugging mld ................................................................. 22
  - debugging mld proxying ...................................................... 27

MLD snooping debugging commands
  - debugging mld-snooping ...................................................... 31

Modem debugging commands
  - debugging modem ......................................................... 36

Monitor Link debugging commands
  - debugging monitor-link ..................................................... 40

MPLS L2VPN debugging commands
  - debugging mpls l2vpn ....................................................... 41

MPLS L3VPN debugging commands
  - debugging l3vpn ........................................................... 50

MPLS OAM debugging commands
  - debugging mpls oam ....................................................... 55

MPLS TE debugging commands
  - debugging isis traffic-eng ............................................... 62
  - debugging mpls moam packet ............................................ 62
  - debugging mpls rsvp-te .................................................... 63
  - debugging mpls te csf ...................................................... 69
  - debugging mpls te management .......................................... 75
  - debugging mpls te protect-switch ..................................... 78
  - debugging ospf mpls-te ................................................... 79

MSDP debugging commands
  - debugging msdp ............................................................ 81

MTR debugging commands
  - debugging multiple-topology event ..................................... 84
  - debugging multiple-topology hsb ....................................... 84

Multicast routing and forwarding debugging commands
  - debugging mlib .............................................................. 86
  - debugging mrm ............................................................. 94
  - debugging mtracert ....................................................... 96
Multicast VPN debugging commands ................................................................. 98
  debugging md ................................................................. 98
  debugging mtunnel .......................................................... 101

MVRP debugging commands ............................................................................. 107
  debugging mvrp error .......................................................... 107
  debugging mvrp event ......................................................... 107
  debugging mvrp packet ....................................................... 108
  debugging mvrp state ........................................................ 109

NAT debugging commands .............................................................................. 113
  debugging nat ................................................................. 113
  debugging userlog nat ...................................................... 116

NAT-PT debugging commands ......................................................................... 118
  debugging natpt .............................................................. 118

ND attack defense debugging commands ......................................................... 121
  debugging ipv6 nd detection packet ........................................ 121

NetStream debugging commands .................................................................... 123
  debugging ip netstream event ............................................. 123
  debugging ip netstream packet ........................................... 128

NQA debugging commands ............................................................................ 130
  debugging nqa ................................................................. 130

NTP debugging commands ............................................................................ 133
  debugging ntp-service ...................................................... 133

OAA debugging command .............................................................................. 139
  debugging oaa remote-interface .......................................... 139

OSPF debugging commands ........................................................................... 140
  debugging ospf event ....................................................... 140
  debugging ospf hot-standby .............................................. 143
  debugging ospf lsa ........................................................... 145
  debugging ospf mpls-te ..................................................... 149
  debugging ospf non-stop-routing ....................................... 152
  debugging ospf packet ..................................................... 155
  debugging ospf spf .......................................................... 160
  debugging ospf timer ....................................................... 176

OSPFv3 debugging commands ....................................................................... 178
  debugging ospfv3 event .................................................... 178
  debugging ospfv3 graceful-restart ..................................... 179
  debugging ospfv3 ifsme ................................................... 182
  debugging ospfv3 lsa ......................................................... 184
  debugging ospfv3 nifsme .................................................. 187
  debugging ospfv3 packet .................................................. 190
  debugging ospfv3 route .................................................... 192

PIM debugging commands .............................................................................. 196
  debugging pim ................................................................. 196

PKI debugging commands ............................................................................. 208
  debugging pki ................................................................. 208

Policy-based routing debugging commands ..................................................... 222
  debugging ip policy-based-route ........................................ 222
RPR debugging commands ................................................................. 308
  debugging rpr error ................................................................. 308
  debugging rpr event ............................................................... 309
  debugging rpr fsm ................................................................. 312
  debugging rpr packet ............................................................. 314
  debugging rpr timer ............................................................... 316
RRPP debugging commands ............................................................. 318
  debugging rrpp ................................................................. 318
RSH debugging commands ............................................................. 326
  debugging rsh ................................................................. 326
Service loopback group debugging commands .................................... 327
  debugging service-loopback .................................................. 327
Session management debugging commands .......................................... 328
  debugging session aging-process ............................................ 328
  debugging session engine ..................................................... 329
  debugging session ext-info ................................................... 332
  debugging session packet-process ......................................... 334
  debugging session relation ................................................... 334
  debugging session session-table ........................................... 336
  debugging session log proc .................................................. 337
  debugging session log active-flow ........................................... 339
SIP debugging commands ............................................................... 341
  debugging voice sip ........................................................... 341
  debugging voice ssm call ..................................................... 353
  debugging voice ssm lsm ..................................................... 363
  debugging voice ssm probe ................................................... 364
  debugging voice ssm reg ..................................................... 367
  debugging voice ssm stack ................................................... 370
  debugging voice ssm sub ..................................................... 377
  debugging voice ssm ucm .................................................... 384
  debugging voice srtpe ........................................................ 385
Smart Link debugging commands ..................................................... 387
  debugging smart-link ......................................................... 387
SNMP debugging commands ........................................................... 391
  debugging snmp agent packet ............................................... 391
  debugging snmp agent process .............................................. 395
  debugging snmp trap packet ................................................ 397
  debugging snmp trap process ............................................... 399
Spanning tree debugging commands .................................................. 402
  debugging stp all ............................................................ 402
  debugging stp event ........................................................ 402
  debugging stp fsm .......................................................... 403
  debugging stp global-error ................................................ 405
  debugging stp global-event ................................................. 407
  debugging stp packet ....................................................... 408
  debugging stp roles ........................................................ 411
  debugging stp tc ............................................................ 412
SSH debugging commands ............................................................. 414
  debugging ssh server ......................................................... 414
<table>
<thead>
<tr>
<th>Command Type</th>
<th>Section Numbers</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRRP debugging commands</td>
<td>SSL debugging commands</td>
<td>515</td>
</tr>
<tr>
<td>SSL debugging commands</td>
<td>SSL debugging commands</td>
<td>424</td>
</tr>
<tr>
<td>SSL VPN debugging commands</td>
<td>SSL VPN debugging commands</td>
<td>431</td>
</tr>
<tr>
<td>Stateful failover debugging commands</td>
<td>SSL VPN debugging commands</td>
<td>433</td>
</tr>
<tr>
<td>Terminal access debugging commands</td>
<td>Terminal access debugging commands</td>
<td>442</td>
</tr>
<tr>
<td>Track debugging commands</td>
<td>Track debugging commands</td>
<td>447</td>
</tr>
<tr>
<td>Tunneling debugging commands</td>
<td>Tunneling debugging commands</td>
<td>450</td>
</tr>
<tr>
<td>UDP helper debugging commands</td>
<td>UDP helper debugging commands</td>
<td>459</td>
</tr>
<tr>
<td>URPF debugging commands</td>
<td>URPF debugging commands</td>
<td>462</td>
</tr>
<tr>
<td>VLAN termination debugging commands</td>
<td>VLAN termination debugging commands</td>
<td>463</td>
</tr>
<tr>
<td>VoFR debugging commands</td>
<td>VoFR debugging commands</td>
<td>467</td>
</tr>
<tr>
<td>Voice subscriber line debugging commands</td>
<td>Voice subscriber line debugging commands</td>
<td>488</td>
</tr>
<tr>
<td>VRRP debugging commands</td>
<td>VRRP debugging commands</td>
<td>515</td>
</tr>
<tr>
<td>IPv4 VRRP debugging commands</td>
<td>IPv4 VRRP debugging commands</td>
<td>515</td>
</tr>
<tr>
<td>IPv6 VRRP debugging commands</td>
<td>IPv6 VRRP debugging commands</td>
<td>523</td>
</tr>
<tr>
<td>WEB filtering debugging commands</td>
<td>WEB filtering debugging commands</td>
<td>533</td>
</tr>
</tbody>
</table>
MAC authentication debugging commands

The MAC authentication module name is identified as “MACAUTH” in debugging messages.

Some information in this chapter is device type specific. Devices in this chapter are categorized depending on their IRF capability and support for interface cards that use independent processors for forwarding traffic, as shown in Table 1.

Table 1 Device types

<table>
<thead>
<tr>
<th>Device type</th>
<th>Interface cards with on-card processors</th>
<th>IRF capability</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed devices</td>
<td>Yes</td>
<td>No</td>
<td>HP 6600 routers (except for 6602)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes (in standalone mode)</td>
<td>HP 12500 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 10500 switches</td>
</tr>
<tr>
<td>Distributed IRF devices</td>
<td>Yes</td>
<td>Yes (in IRF mode)</td>
<td>HP 12500 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 10500 switches</td>
</tr>
<tr>
<td>Centralized devices</td>
<td>No</td>
<td>No</td>
<td>HP MSR routers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 6602 router</td>
</tr>
<tr>
<td>Centralized IRF devices</td>
<td>No</td>
<td>Yes</td>
<td>HP 5800 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 5500 switches</td>
</tr>
</tbody>
</table>

The output description tables in this document only contain fields and messages that require an explanation.

debugging mac-authentication

Use `debugging mac-authentication` to enable MAC authentication debugging.

Use `undo debugging mac-authentication` to disable MAC authentication debugging.

Syntax

Centralized devices:

`debugging mac-authentication event`

`undo debugging mac-authentication event`

Distributed devices/centralized IRF devices:

`debugging mac-authentication event [ slot slot-number ]`

`undo debugging mac-authentication event [ slot slot-number ]`

Distributed IRF devices:

`debugging mac-authentication event [ chassis chassis-number slot slot-number ]`

`undo debugging mac-authentication event [ chassis chassis-number slot slot-number ]`
Default

MAC authentication debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

event: Specifies MAC authentication event debugging.

slot slot-number: Specifies a card by its slot number. (Distributed device—In standalone mode.)

slot slot-number: Specifies an IRF member device by its member ID. (Centralized IRF devices.)

classis chassis-number slot slot-number: Specifies a card on an IRF member device. The chassis-number argument represents the ID of the IRF member device. The slot-number argument represents the slot number of the card. (Distributed devices—In IRF mode.)

Usage guidelines

Table 2 describes output fields and messages for the debugging mac-authentication event command.

Table 2 Output from the debugging mac-authentication event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IfIndex : IfIndex</td>
<td>Interface index.</td>
</tr>
<tr>
<td>cmd = CmdType</td>
<td>Command.</td>
</tr>
<tr>
<td>Processing node CONNECTING...</td>
<td>The MAC authentication module is processing the node in Connecting state.</td>
</tr>
<tr>
<td>Processing node SUCCESS...</td>
<td>The MAC authentication module is processing the node in Success state.</td>
</tr>
<tr>
<td>Processing node FAILURE...</td>
<td>The MAC authentication module is processing the node in FAILURE state.</td>
</tr>
<tr>
<td>Processing node LOGOFF...</td>
<td>The MAC authentication module is processing the node in LOGOFF state.</td>
</tr>
<tr>
<td>Warning:MACAuth Que is almost full, CUT user msg is dropped</td>
<td>MAC authentication message queue is to be full. The cut user connection messages will be dropped.</td>
</tr>
<tr>
<td>new mac address MacAddress</td>
<td>The MAC authentication module received unknown MAC address from the underlayer.</td>
</tr>
<tr>
<td>PortIndex is invalid .</td>
<td>Invalid port index.</td>
</tr>
<tr>
<td>error mac address MacAddress</td>
<td>Invalid MAC address.</td>
</tr>
<tr>
<td>Send new mac event to queue failed .</td>
<td>The MAC authentication module failed to send new MAC messages to the message queue.</td>
</tr>
<tr>
<td>write new mac event failed</td>
<td>The MAC authentication module failed to write new MAC addresses.</td>
</tr>
<tr>
<td>MAC_AddressLearn rcv a multicast MAC addr , will not authenticate...</td>
<td>The device received a multicast MAC address. MAC authentication does not authenticate the multicast address.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MacAuth is configuring or already disabled ,can't add src drop mac .</td>
<td>MAC authentication is under configuration or has been disabled. The device cannot add the MAC address to the MAC address table that contains blocked MAC addresses.</td>
</tr>
<tr>
<td>Broadcast send a MAC address syn add msg</td>
<td>The device broadcast the synchronization message for a new MAC.</td>
</tr>
<tr>
<td>Broadcast send a new slot inserted notify msg ,new slot is : SlotID</td>
<td>The device broadcast the message to notify the insertion of a new card. (Distributed devices/distributed IRF devices.)</td>
</tr>
<tr>
<td>Receive broadcast kickoff preUser IPC request.</td>
<td>The MAC authentication module received broadcast IPC request to remove duplicated MAC authentication users.</td>
</tr>
<tr>
<td>find auth by index fail...</td>
<td>The device failed to find MAC authentication resources based on the user index.</td>
</tr>
<tr>
<td>NOT FOUND MacAddress, Vlan: VlanID in MACAUTH blocked List.</td>
<td>The MAC authentication module failed to find the MAC address of the specified VLAN in the quiet MAC address table.</td>
</tr>
<tr>
<td>FOUND MacAddress, Vlan: VlanID in MACAUTH blocked List.</td>
<td>The MAC authentication module found the MAC addresses of the specified VLAN in the quiet MAC address table.</td>
</tr>
<tr>
<td>Send request to driver to get port PortSec info for IfIndex = IfIndex, cmd = CmdType</td>
<td>The MAC authentication module sent a request to the driver. The module requested port security information of a port.</td>
</tr>
<tr>
<td>Get Port Portsec info for IfIndex = IfIndex, cmd = CmdType, return code = ErrCode</td>
<td>The MAC authentication module received port security information of the port.</td>
</tr>
<tr>
<td>IPCMsg: Sync Slot Insert Information msg: recv data is null!</td>
<td>Card insertion synchronization message. The received data is null. (Distributed devices/distributed IRF devices.)</td>
</tr>
<tr>
<td>IPCMsg: Sync Slot Insert Information msg: recv data length is wrong!</td>
<td>Card insertion synchronization message. The received data length is incorrect. (Distributed devices.)</td>
</tr>
<tr>
<td></td>
<td>Member device insertion synchronization message. The received data is null. (Centralized IRF devices.)</td>
</tr>
<tr>
<td></td>
<td>Member device insertion synchronization message. The received data length is incorrect. (Centralized IRF devices.)</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SlotInsert, Portsec not enabled, macauth enabled, enable macauth on new slot!</td>
<td>When a new card was inserted, port security was not enabled and MAC authentication was enabled on the card. After the insertion, MAC authentication is enabled on the new card. (Distributed devices.)</td>
</tr>
<tr>
<td></td>
<td>When a new member device was inserted, port security was not enabled and MAC authentication was enabled on the device. After the insertion, MAC authentication is enabled on the new member device. (Centralized IRF devices.)</td>
</tr>
<tr>
<td>SlotInsert, Portsec enabled or macauth not enabled, enable macauth on new slot!</td>
<td>When a new card was inserted, port security was enabled or MAC authentication was not enabled on the card. After the insertion, MAC authentication is enabled on the new card. (Distributed devices.)</td>
</tr>
<tr>
<td></td>
<td>When a new member device was inserted, port security was enabled or MAC authentication was not enabled on the device. After the insertion, MAC authentication is enabled on the new member device. (Centralized IRF devices.)</td>
</tr>
<tr>
<td>Fail to enable mac_auth for driver’s err</td>
<td>The system failed to enable MAC authentication because of driver error.</td>
</tr>
<tr>
<td>Fail to disable mac_auth for driver’s err</td>
<td>The system failed to disable MAC authentication because of driver error.</td>
</tr>
<tr>
<td>Receive reset drv msg</td>
<td>The MAC authentication module received a message to reset the driver.</td>
</tr>
<tr>
<td>Receive clear drv msg</td>
<td>The MAC authentication module received a message to clear the driver.</td>
</tr>
<tr>
<td>waiting users off-line</td>
<td>The MAC authentication module is waiting for users to go offline.</td>
</tr>
<tr>
<td>clear mac-auth drv</td>
<td>The MAC authentication module cleared the MAC authentication driver.</td>
</tr>
<tr>
<td>Port leave from Vlan, but user is leaving GuestVLAN, need NOT cut user!!</td>
<td>The interface has left the guest VLAN, and the user is being removed from the guest VLAN. In this case, it is not required to force the user to go offline.</td>
</tr>
<tr>
<td>Port leave from Vlan, cut user!!</td>
<td>The interface has left guest VLAN. The user went offline forcibly.</td>
</tr>
</tbody>
</table>
| Added an MACAUTH MGV entry for MacAddress: IfIndex =IfIndex Guest-Vlan=VlanID | An MGV entry for MAC authentication was added:  
  - **MacAddress**—MAC address added to the MGV entry.  
  - **IfIndex**—Port index.  
  - **Guest-Vlan**—Guest VLAN ID.                                                                                                                                                                                                                                                                |
| MGV traffic timer triggered MAC authentication: MAC= MacAddress , IfIndex =IfIndex , Guest-Vlan=VlanID, InitVlan = InitVlan | MGV traffic timer triggered MAC authentication.  
  - **MAC**—MAC address that triggered the MAC authentication.  
  - **IfIndex**—Port index.  
  - **Guest-Vlan**—Guest VLAN ID.  
  - **InitVlan**—Initial VLAN to which the port was assigned before it was added to the guest VLAN.                                                                                                                                                                                                 |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>An MGV entry has had no packets matching it and therefore needs to be deleted.</td>
<td>No packets match the MGV entry. The MGV entry will be deleted.</td>
</tr>
<tr>
<td>Failed to process MGV for the same MGV entry.</td>
<td>The MAC authentication module failed to process MGV because the same MGV entry already exists.</td>
</tr>
</tbody>
</table>

**Examples**

Enable MAC authentication event debugging. Output similar to the following example is generated under the following conditions:

- MAC authentication is enabled globally and on GigabitEthernet 1/1.
- A user passes MAC authentication on GigabitEthernet 1/1.
- The network cable connected to GigabitEthernet 1/1 is removed.

```plaintext
<Sysname> debugging mac-authentication event
<Sysname> Sysname-view
[Sysname] interface gigabitethernet 1/1
[Sysname-GigabitEthernet1/1] mac-authentication
*Apr 30 12:14:27:357 2000 Sysname MACAUTH/7/EVENT:Port:GigabitEthernet1/1,send request to driver to Get Port Portsec info for IfIndex = 900002,cmd = 8460f01
*Apr 30 12:14:27:369 2000 Sysname MACAUTH/7/EVENT:Port:GigabitEthernet1/1, get Port Portsec info for IfIndex = 900002,cmd = 8460f01,return code = 0
Mac-auth is enabled on port GigabitEthernet1/1.

// The MAC authentication module sent a request to the driver. The request obtained port security information on GigabitEthernet 1/1 and enabled MAC authentication on the port.
```

```plaintext
*Apr 30 12:19:05:199 2000 Sysname MACAUTH/7/EVENT:Port:GigabitEthernet1/1, new mac address 0015-e947-e45b

// GigabitEthernet 1/1 received the authentication request from the user, and the MAC authentication module authenticated the new MAC address.
```

```plaintext
*Apr 30 12:24:48:600 2000 Sysname MACAUTH/7/EVENT:Port:GigabitEthernet1/1, Auth:0,IF_IFMSG_DOWN

// MAC authentication on the port failed after you unplugged the network cable connected to the port.
```
MAC-in-MAC debugging commands

The MAC-in-MAC module name is identified as "MINM" in debugging messages. The output description tables in this document only contain fields and messages that require an explanation.

debugging minm

Use debugging minm to enable debugging for MAC-in-MAC. Use undo debugging minm to disable debugging for MAC-in-MAC.

Syntax

debugging minm { all | error | event | packet }
undo debugging minm { all | error | event | packet }

Default

Debugging for MAC-in-MAC is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all: All types of debugging for MAC-in-MAC.
error: MAC-in-MAC error debugging.
event: MAC-in-MAC event debugging.
packet: MAC-in-MAC frame debugging.

Usage guidelines

Table 1 describes the output fields and messages for the debugging minm error command.

Table 3 Output from the debugging minm error command

<table>
<thead>
<tr>
<th>a. Field</th>
<th>b. Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c. The ISID of this packet is</td>
<td>d. The I-SID of the MAC-in-MAC frame is not as configured.</td>
</tr>
<tr>
<td>not configured</td>
<td></td>
</tr>
<tr>
<td>e. Failed to set driver</td>
<td>f. MINM failed to issue the configuration to the driver.</td>
</tr>
<tr>
<td>g. Failed to process BMAC aging</td>
<td>h. MINM failed to age the B-MAC entry.</td>
</tr>
<tr>
<td>event</td>
<td></td>
</tr>
<tr>
<td>i. Failed to create VSI, when</td>
<td>j. MINM failed to create a VSI when a board was plugged.</td>
</tr>
<tr>
<td>processing slot in</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>k. Failed to set driver type: MINM_DEL</td>
<td>l. MINM failed to issue MINM_DEL to the driver.</td>
</tr>
<tr>
<td>m. Interface type is not supported</td>
<td>n. This type of interface does not support MAC-in-MAC.</td>
</tr>
<tr>
<td>o. Failed to broadcast message by VPLS IPC</td>
<td>p. MINM failed to broadcast the IPC message through VPLS.</td>
</tr>
</tbody>
</table>

Table 2 describes the output fields and messages for the **debugging minm event** command.
Table 4 Output from the debugging minm event command

<table>
<thead>
<tr>
<th>a. Field</th>
<th>b. Description</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d.</td>
<td>MINM succeeded in processing IPC messages X. X indicates the IPC message type:</td>
</tr>
<tr>
<td>0</td>
<td>Invalid operation type.</td>
</tr>
<tr>
<td>1</td>
<td>Adding VSI.</td>
</tr>
<tr>
<td>2</td>
<td>Deleting VSI.</td>
</tr>
<tr>
<td>3</td>
<td>Configuring B-VLAN.</td>
</tr>
<tr>
<td>4</td>
<td>Adding B-VLAN to the driver.</td>
</tr>
<tr>
<td>5</td>
<td>Deleting B-VLAN to the driver.</td>
</tr>
<tr>
<td>6</td>
<td>Deleting uplink port to the driver.</td>
</tr>
<tr>
<td>7</td>
<td>Adding uplink port to the driver.</td>
</tr>
<tr>
<td>8</td>
<td>Deleting uplink port.</td>
</tr>
<tr>
<td>9</td>
<td>Deleting an uplink port by the port number.</td>
</tr>
<tr>
<td>10</td>
<td>Adding uplink port.</td>
</tr>
<tr>
<td>11</td>
<td>Enabling re-encapsulation.</td>
</tr>
<tr>
<td>12</td>
<td>Disabling re-encapsulation.</td>
</tr>
<tr>
<td>13</td>
<td>Adding B-MAC.</td>
</tr>
<tr>
<td>14</td>
<td>Modifying B-MAC.</td>
</tr>
<tr>
<td>15</td>
<td>Deleting all B-MACs.</td>
</tr>
<tr>
<td>16</td>
<td>Deleting B-MACs of the specified VSI.</td>
</tr>
<tr>
<td>17</td>
<td>Deleting B-MACs of the specified VSI and link ID.</td>
</tr>
<tr>
<td>18</td>
<td>Deleting B-MACs of the specified VLAN.</td>
</tr>
<tr>
<td>19</td>
<td>Deleting B-MACs of the specified port.</td>
</tr>
<tr>
<td>20</td>
<td>Deleting B-MACs of the specified VLAN and port.</td>
</tr>
<tr>
<td>21</td>
<td>Deleting B-MAC by MAC.</td>
</tr>
<tr>
<td>22</td>
<td>Aging B-MACs of the specified VSI.</td>
</tr>
<tr>
<td>23</td>
<td>Aging B-MACs of the specified link ID.</td>
</tr>
<tr>
<td>24</td>
<td>Debugging.</td>
</tr>
<tr>
<td>25</td>
<td>Setting board boot.</td>
</tr>
<tr>
<td>26</td>
<td>Setting state for all boards.</td>
</tr>
<tr>
<td>27</td>
<td>Setting state for the VSI.</td>
</tr>
<tr>
<td>28</td>
<td>Processing B-MAC information according to the B-MAC state.</td>
</tr>
<tr>
<td>29</td>
<td>Displaying data.</td>
</tr>
<tr>
<td>30</td>
<td>Configuring VSI when a board was plugged.</td>
</tr>
<tr>
<td>31</td>
<td>Configuring uplink port when a board was plugged.</td>
</tr>
</tbody>
</table>

**c.** Succeeded in processing IPC message: X
<table>
<thead>
<tr>
<th>a. Field</th>
<th>b. Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>e. Succeeded in sending IPC to slot ( n ), message type: ( X )</td>
<td>f. MINM succeeded in sending IPC message ( X ) to the board in slot ( n ). ( n ) indicates the slot number of the board. ( X ) indicates the IPC message type.</td>
</tr>
<tr>
<td>g. Succeeded in processing interface ( n ), Event: ( Y )</td>
<td>h. MINM succeeded in processing event ( Y ) on port ( n ). ( n ) indicates the number of the port. ( Y ) indicates the event type:</td>
</tr>
</tbody>
</table>
|   - 2—Deleting a port.  
   - 8—Unplugging a port.  
   - 64—Port down. | |
| i. Succeeded in processing VSI event: \( Y \) | j. MINM succeeded in processing VSI event \( Y \). \( Y \) indicates the event type: |
|   - 0—Creating VSI.  
   - 1—Deleting VSI.  
   - 2—Shutting down VSI.  
   - 3—Bringing up VSI. | |
| k. Succeeded in processing event: \( Y \). \( u\text{StartIndex} = a \), \( u\text{EndIndex} = b \), \( u\text{Data} = c \) | l. MINM succeeded in processing VLAN event \( Y \), with the start address \( a \), end address \( b \), and data \( c \). \( Y \) indicates the event type: |
|   - 4—Static VLAN deleting detection.  
   - 2048—Deleting an access port from the VLAN.  
   - 8192—Deleting a trunk port from the VLAN.  
   - 32768—Deleting a multi port from the VLAN.  
   - 2097152—Deleting ports in batch from the VLAN.  
   - 8388608—Deleting a port from multiple VLANs in batch. | |
| m. Succeeded in processing slot \( n \) event: SLOT_IN | n. MINM succeeded in processing the event of plugging a board into slot \( n \). \( n \) is the slot number of the board. |
### Table 5: Output from the debugging minm packet command

<table>
<thead>
<tr>
<th>a. Field</th>
<th>b. Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c. Receive MAC-in-MAC frame from <code>IFname</code></td>
<td>d. MINM received MAC-in-MAC frames on port <code>IFname</code>.</td>
</tr>
<tr>
<td>e. B-DA</td>
<td>f. Destination B-MAC address.</td>
</tr>
<tr>
<td>g. B-SA</td>
<td>h. Source B-MAC address.</td>
</tr>
<tr>
<td>i. BTAG TPID</td>
<td>j. BTAG type, which is fixed to 0x88a8.</td>
</tr>
<tr>
<td>k. ITAG TPID</td>
<td>l. ITAG type, which is fixed to 0x88e7.</td>
</tr>
<tr>
<td>m. C-DA</td>
<td>n. Destination C-MAC address.</td>
</tr>
<tr>
<td>o. C-SA</td>
<td>p. Source C-MAC address.</td>
</tr>
</tbody>
</table>

### Examples

#### # Enable MAC-in-MAC error debugging. Output similar to the following example is generated when you configure the uplink port and downlink port for MAC-in-MAC.

```plaintext
<Sysname> debugging minm error
*Oct 24 14:23:58:531 2009 Sysname MINM/7/Error:  This packet is not received from BVLAN 1.
*Oct 24 14:23:58:531 2009 Sysname MINM/7/Error:  Failed to learn BMAC.
// An uplink port in B-VLAN 1 received MAC-in-MAC frames from B-VLAN 20, and B-MAC learning failed.
```

#### # Enable MAC-in-MAC event debugging. Output similar to the following example is generated when you configure the uplink port and downlink port for MAC-in-MAC.

```plaintext
<Sysname> debugging minm event
// MINM succeeded in sending IPC message 13 (adding B-MAC) to the board in slot ffffffff.
*Oct 24 14:34:27:515 2009 Quidway MINM/7/Event:  Succeeded in processing BMAC aging event
// MINM succeeded in aging the B-MAC entry.
*Oct 24 14:36:35:312 2009 Quidway MINM/7/Event:  Succeeded in processing VSI event: 2
// MINM succeeded in processing VSI event 2 (shutting down VSI).
```

// MINM already learned the B-MAC.

# Enable MAC-in-MAC packet debugging. Output similar to the following example is generated when you configure the uplink port and downlink port for MAC-in-MAC.

<Sysname> debugging minm packet

*Oct 24 11:20:41:453 2009 Sysname MINM/7/Packet:

Receive MAC-in-MAC frame from Ethernet1/1

   B-DA: 0102-0304-0506
   B-SA: 0605-0403-0206
   BTAG TPID: 0x88a8
   BVLAN: 20
   ITAG TPID: 0x88e7
   I-SID: 111
   C-DA: 0101-0101-0101
   C-SA: 0202-0202-0202

// Content of the MAC-in-MAC frame received from port Ethernet 1/1.
MBGP debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging bgp update ipv4 multicast

Use `debugging bgp update ipv4 multicast` to enable BGP update debugging for IPv4-MBGP address family.

Use `undo debugging bgp update ipv4` to disable BGP update debugging for IPv4-MBGP address family.

Syntax

```
debugging bgp update ipv4 multicast [ peer { ip-address | group-name } ] [ receive | send ] [ verbose ]
undo debugging bgp update ipv4 multicast [ peer { ip-address | group-name } ] [ receive | send ] [ verbose ]
```

Default

BGP update debugging for IPv4-MBGP address family is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- **ip-address**: Specifies the IP address of the peer entity.
- **group-name**: Specifies the peer group, which is a string of 1 to 47 characters.
- **receive**: Specifies debugging for received BGP updates.
- **send**: Specifies debugging for sent BGP updates.
- **verbose**: Displays detailed debugging information.

Usage guidelines

1. **IMPORTANT:**
   
   Enabling debugging might affect system performance. Use this command only when necessary and disable debugging when the debugging operation is complete.

   Table 1 describes the output fields and messages for the `debugging bgp update ipv4 multicast` command.

   **Table 6 Output from the debugging bgp update ipv4 multicast command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP.xxx</td>
<td>Name of the current instance.</td>
</tr>
<tr>
<td>Recv UPDATE from x.x.x.x</td>
<td>BGP received an update from BGP neighbor x.x.x.x.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Recv UPDATE(Withdraw) from x.x.x.x</td>
<td>BGP received an update withdraw from BGP neighbor x.x.x.x.</td>
</tr>
<tr>
<td>Error identified while receiving UPDATE message from the peer x.x.x.x and ignored</td>
<td>Error was found in the received update message from neighbor x.x.x.x, and the message was ignored.</td>
</tr>
<tr>
<td>Err/SubEr</td>
<td>Error code/sub-error code.</td>
</tr>
<tr>
<td>Errdata:</td>
<td>Error data.</td>
</tr>
<tr>
<td>Send UPDATE to x.x.x.x</td>
<td>BGP sent an update message to BGP neighbor x.x.x.x.</td>
</tr>
<tr>
<td>Send UPDATE(Withdraw) to peer x.x.x.x</td>
<td>BGP sent an update withdraw message to BGP neighbor x.x.x.x.</td>
</tr>
<tr>
<td>x.x.x.x/xx</td>
<td>Destination IP address/mask.</td>
</tr>
<tr>
<td>INBOUND</td>
<td>Inbound policy.</td>
</tr>
<tr>
<td>LocRemCross:Export Policy Check Failed</td>
<td>BGP exported policy check error when importing routes to VRF routing table.</td>
</tr>
<tr>
<td>LocRemCross:Import Policy Check Error</td>
<td>BGP imported policy check error when importing routes to VRF routing table.</td>
</tr>
<tr>
<td>in-label</td>
<td>Incoming label.</td>
</tr>
<tr>
<td>out-label</td>
<td>Outgoing label.</td>
</tr>
<tr>
<td>tnl id</td>
<td>Tunnel ID.</td>
</tr>
<tr>
<td>Create ILM Success</td>
<td>BGP succeeded to create the ILM.</td>
</tr>
<tr>
<td>Delete ILM Success</td>
<td>BGP succeeded to delete the ILM.</td>
</tr>
<tr>
<td>Create NHLFE success</td>
<td>BGP succeeded to create the NHLFE.</td>
</tr>
<tr>
<td>Delete NHLFE success</td>
<td>BGP succeeded to delete the NHLFE.</td>
</tr>
<tr>
<td>afi = 196(l2vpn) safi = 128(l2vpn)</td>
<td>The address family was 196, and the sub-address family was 128(L2VPN).</td>
</tr>
<tr>
<td>Origin</td>
<td>Origin of BGP.</td>
</tr>
<tr>
<td>AS Path</td>
<td>AS Path of BGP.</td>
</tr>
<tr>
<td>Next Hop</td>
<td>Next Hop of BGP.</td>
</tr>
<tr>
<td>Local Pref</td>
<td>Local Preference of BGP.</td>
</tr>
<tr>
<td>MED</td>
<td>MED attribute of BGP.</td>
</tr>
<tr>
<td>afi = 155(vpls) safi = 128(vpls)</td>
<td>The address family was 155, and the sub-address family was 128(VPLS).</td>
</tr>
<tr>
<td>Create ILM6 Success</td>
<td>BGP succeeded to create the ILM6.</td>
</tr>
<tr>
<td>Delete ILM6 Success</td>
<td>BGP succeeded to delete the ILM6.</td>
</tr>
<tr>
<td>BGP Recv Label Route : Tunnel Waiting , BGP LSP isn't created</td>
<td>BGP received a labeled route, but the tunnel was still waiting, and the BGP LSP had not been established.</td>
</tr>
<tr>
<td>RouteDistinguisher:X:X</td>
<td>RD value.</td>
</tr>
<tr>
<td>label offset:X</td>
<td>Label offset.</td>
</tr>
<tr>
<td>label base:XXX</td>
<td>Label start position.</td>
</tr>
</tbody>
</table>
# Enable BGP update debugging for IPv4-MBGP address family.

<Sysname> debugging bgp update ipv4 multicast packet

*Apr 2 17:46:55:830 2007 Sysname-Comware RM/6/RMDEBUG:

BGP_MBGP.: Recv UPDATE from 12.1.1.1 with following destinations :

Update message length : 67
Local Pref   : 100
MED          : 0
Origin       : Incomplete
AS Path      :
Next Hop     : 12.1.1.1
1.2.3.4/32
MFF debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging mac-forced-forwarding

Use debugging mac-forced-forwarding to enable MFF debugging.

Use undo debugging mac-forced-forwarding to disable MFF debugging.

Syntax

debugging mac-forced-forwarding { all | error | event | info | packet }
undo debugging mac-forced-forwarding { all | error | event | info | packet }

Default

MFF debugging is disabled.

Views

User view

Default command level

2: System level

Parameters

all: Specifies all types of debugging for MFF.
error: Specifies error debugging.
event: Specifies error debugging.
info: Specifies information debugging.
packet: Specifies packet debugging.

Usage guidelines

Table 1 describes output fields and messages for the debugging mac-forced-forwarding error command.

Table 7 Output from the debugging mac-forced-forwarding error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to fill vlan tag</td>
<td>MFF failed to fill the VLAN tag when constructing an ARP packet.</td>
</tr>
<tr>
<td>Failed to cut vlan tag</td>
<td>MFF failed to remove the VLAN tag when constructing an ARP packet.</td>
</tr>
<tr>
<td>Failed to set gateway</td>
<td>MFF failed to update the gateway MAC address.</td>
</tr>
<tr>
<td>Failed to get gateway</td>
<td>MFF failed to find the user’s gateway in the VLAN.</td>
</tr>
<tr>
<td>CFA send failed</td>
<td>MFF failed to send IPC broadcast.</td>
</tr>
<tr>
<td>Replace buildrun failed</td>
<td>MFF failed to replace buildrun.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>Process event of slot-insert failed</td>
<td>MFF failed to process a card insertion event.</td>
</tr>
<tr>
<td>Create interface control block failed</td>
<td>MFF failed to create an interface control block.</td>
</tr>
<tr>
<td>Delete interface control block failed</td>
<td>MFF failed to delete an interface control block.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the **debugging mac-forced-forwarding event** command.

### Table 8: Output from the debugging mac-forced-forwarding event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deal MAC change</td>
<td>The driver notified MFF of a MAC address change.</td>
</tr>
<tr>
<td>Notify MCM ACL rule</td>
<td>MFF notified MCM to apply or delete the ACL.</td>
</tr>
<tr>
<td>Interface name:  \textit{ifName}</td>
<td>Name of the interface to which the ACL applies, action for the ACL (add or delete), and the returned result.</td>
</tr>
<tr>
<td>Option type: \textit{optionType} Return Value: \textit{n}</td>
<td></td>
</tr>
<tr>
<td>Notify DHSP ACL rule</td>
<td>MFF notified DHCP snooping to apply or delete the ACL.</td>
</tr>
<tr>
<td>Interface name:  \textit{ifName}, \textit{portType},</td>
<td>Name of the interface to which the ACL applies, action for the ACL (add or delete), and the returned result.</td>
</tr>
<tr>
<td>Option type: \textit{optionType} Return Value: \textit{n}</td>
<td></td>
</tr>
<tr>
<td>Notify DRV ACL rule</td>
<td>MFF notified driver to apply or delete the ACL.</td>
</tr>
<tr>
<td>Interface name:  \textit{ifName}</td>
<td>Name of the interface to which the ACL applies, and VLAN ID of the interface.</td>
</tr>
<tr>
<td>Port type: \textit{portType} Option</td>
<td>Interface type, and action for the ACL (add or delete).</td>
</tr>
<tr>
<td>Rule type: \textit{ruleType}, Return Value: \textit{n}</td>
<td>ACL type, and returned result.</td>
</tr>
<tr>
<td>Notify ISG ACL rule</td>
<td>MFF notified IP source guard to apply or delete the ACL.</td>
</tr>
<tr>
<td>Interface name:  \textit{ifName}</td>
<td>Name of the interface to which the ACL applies.</td>
</tr>
<tr>
<td>VlanId: \textit{vlanId}</td>
<td>VLAN ID of the interface.</td>
</tr>
<tr>
<td>Option type: \textit{optionType} Gateway IP: \textit{x.x.x.x}</td>
<td>Action for the ACL (add or delete), and gateway IP address.</td>
</tr>
<tr>
<td>Gateway MAC: \textit{xxxx:xxxx:xxxx:xxxx} Return Value: \textit{n}</td>
<td>Gateway MAC address, and returned result.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the **debugging mac-forced-forwarding info** command.

### Table 9: Output from the debugging mac-forced-forwarding info command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify other application MFF is enabled</td>
<td>MFF notified other modules that MFF was enabled.</td>
</tr>
<tr>
<td>Notify other application MFF is disabled</td>
<td>MFF notified other modules that MFF was disabled.</td>
</tr>
<tr>
<td>No gateway in VLAN \textit{n}</td>
<td>No gateway was found for the ARP request received by MFF.</td>
</tr>
<tr>
<td>Destination port index can’t found</td>
<td>No outbound interface was not found when MFF sent packets.</td>
</tr>
<tr>
<td>Transmit request from gateway</td>
<td>After the network port received an ARP request for an unknown host from the gateway, MFF forwarded the request.</td>
</tr>
<tr>
<td>Transmit request from server</td>
<td>After the network port received an ARP request for an unknown host from the server, MFF forwarded the request.</td>
</tr>
</tbody>
</table>

17
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit request from client</td>
<td>After the user received an ARP request from a host and the MAC address of the gateway was not obtained, MFF forwarded the request.</td>
</tr>
<tr>
<td>Refresh gateway information by packet</td>
<td>MFF updated the MAC address of the gateway and the upstream network port marked by the gateway.</td>
</tr>
<tr>
<td>Self ip address</td>
<td>The MFF device did not process the packet with the destination IP address as that of a virtual interface on itself.</td>
</tr>
</tbody>
</table>

### Examples

```
# Enable MFF event debugging on an MFF enabled device.
<Sysname> debugging mac-forced-forwarding event
<Sysname> terminal debugging
% Current terminal debugging is on

// Event debugging was enabled.
*Apr 26 16:30:21:435 2000 Sysname MFF/7/Debug_MFF_Event:
  MFF Event:
    Notify DHCP ACL rule:
      Interface name: Ethernet1/3,     Port type: USER,
      Option type: ADD,       Return Value: 0

// After automatic mode MFF was enabled for VLAN 100, MFF notified the DHCP module to apply an ACL to user port Ethernet 1/3 to permit packets from DHCP clients. The apply operation was successful.
*Apr 26 16:30:21:441 2000 Sysname MFF/7/Debug_MFF_Event:
  MFF Event:
    Notify DRV ACL rule:
      Interface name: Ethernet1/3,     VlanId: 100,
      Port type: USER,       Option type: ADD,
      Rule type: ALL DENY,       Return Value: 0

// After automatic mode MFF was enabled for VLAN 100, MFF notified the driver to apply an ACL to user port Ethernet 1/3 to deny all packets. The apply operation was successful.
*Apr 26 16:30:21:451 2000 Sysname MFF/7/Debug_MFF_Event:
  MFF Event:
    Notify DRV ACL rule:
      Interface name: Ethernet1/3,     VlanId: 100,
      Port type: USER,       Option type: ADD,
      Rule type: MULTICAST PERMIT,  Return Value: 0

// After automatic mode MFF was enabled for VLAN 100, MFF notified the driver to apply an ACL to user port Ethernet 1/3 to permit multicast packets. The apply operation was successful.
*Apr 26 16:30:21:461 2000 Sysname MFF/7/Debug_MFF_Event:
  MFF Event:
    Notify ISG ACL rule:
      Interface name: Ethernet1/3,     VlanId: 100,
      Option type: ENABLE,    Gateway IP: N/A,
      Gateway MAC: N/A    Return Value: 0

// After automatic mode MFF was enabled for VLAN 100, MFF notified ISG to apply an ACL to enable user port Ethernet 1/3, and the result was successful.
```
Apr 26 16:30:21:471 2000 Sysname MFF/7/Debug_MFF_Event:
MFF Event:
Notify MCM ACL rule:
Interface name: Ethernet1/3,
Option type: ADD, Return Value: 0
// After automatic mode MFF was enabled for VLAN 100, MFF notified MCM to apply an ACL to user port Ethernet 1/3 to deliver ARP packets to the CPU. The apply operation was successful.

Apr 26 16:30:21:582 2000 Sysname MFF/7/Debug_MFF_Event:
MFF Event:
Notify DHSP ACL rule:
Interface name: Ethernet1/2, Port type: NETWORK,
Option type: ADD, Return Value: 0
// After automatic mode MFF was enabled for VLAN 100, MFF notified DHCP to apply an ACL to network port Ethernet 1/2 to permit DHCP packets. The apply operation was successful.

Apr 26 16:30:21:592 2000 Sysname MFF/7/Debug_MFF_Event:
MFF Event:
Notify DRV ACL rule:
Interface name: Ethernet1/2, VlanId: 100,
Port type: NETWORK, Option type: ADD,
Rule type: BROADCAST DENY, Return Value: 0
// After automatic mode MFF was enabled for VLAN 100, MFF notified the driver to apply an ACL to network port Ethernet 1/2 to deny multicast packets. The apply operation was successful.

Apr 26 16:30:21:602 2000 Sysname MFF/7/Debug_MFF_Event:
MFF Event:
Notify DRV ACL rule:
Interface name: Ethernet1/2, VlanId: 100,
Port type: NETWORK, Option type: ADD,
Rule type: MULTICAST PERMIT, Return Value: 0
// After automatic mode MFF was enabled for VLAN 100, MFF notified the driver to apply an ACL to network port Ethernet 1/2 to permit multicast packets. The apply operation was successful.

Apr 26 16:30:21:612 2000 Sysname MFF/7/Debug_MFF_Event:
MFF Event:
Notify MCM ACL rule:
Interface name: Ethernet1/2,
Option type: ADD, Return Value: 0
// After automatic mode MFF was enabled for VLAN 100, MFF notified MCM to apply an ACL to network port Ethernet 1/2 to deliver ARP packets to the CPU. The apply operation was successful.

# Enable MFF error debugging on an MFF enabled device.
<Sysname> debugging mac-forced-forwarding error
<Sysname> terminal debugging
% Current terminal debugging is on
// Enable MFF error debugging was enabled.
Error: MAC-forced forwarding is not enabled in the VLAN.
// Periodic gateway MAC probe cannot be enabled because MFF was not enabled.
# Enable MFF information debugging on an MFF enabled device.
<Sysname> debugging mac-forced-forwarding info
<Sysname> terminal debugging

% Current terminal debugging is on

// MFF information debugging was enabled.
*Apr 26 17:00:25:371 2000 Sysname MFF/7/Debug_MFF_Info:
MFF Info:
Get client info from DHCP:
Client MAC: 000d-5619-f7bc,  Client IP: 100.1.1.1,  Client VLAN: 100,
Gateway IP: 100.1.1.100,  Port: Ethernet1/3

// After automatic mode MFF was enabled, MFF obtained a client entry from the DHCP server.
# Enable MFF packet debugging on an MFF enabled device.
<Sysname> debugging mac-forced-forwarding packet
<Sysname> terminal debugging

% Current terminal debugging is on

// MFF packet debugging was enabled.
*Apr 26 12:12:59:465 2000 Sysname MFF/7/Debug_MFF_Recv_Pkt:
Receiving MFF packet:
Src Interface : Ethernet1/2
Dst Interface :  N/A
InterfaceType :NETWORK    Vlan ID :100
SrcMac :00e0-fc00-3102    SrcIp :     100.1.1.100
DstMac :0000-0000-0000    DstIp :       100.1.1.1
PacketType :REQUEST

// Network port Ethernet1/2 received an ARP request for the MAC address of host 100.1.1.1 from the
gateway with IP address 100.1.1.100 and MAC address 00e0-fc00-3102.
*Apr 26 12:12:59:465 2000 Sysname MFF/7/Debug_MFF_Recv_Pkt:
Receiving MFF packet:
Src Interface : Ethernet1/3
Dst Interface :  N/A
InterfaceType :USER    Vlan ID :100
SrcMac :000d-5619-f7bc    SrcIp : 100.1.1.1
DstMac :0000-0000-0000    DstIp : 100.1.1.1
PacketType :GRATUITOUS

// User port Ethernet1/3 received a gratuitous ARP packet from the host with IP address 100.1.1.1 and
MAC address 000d-5619-f7bc.
*Apr 26 12:12:59:509 2000 Sysname MFF/7/Debug_MFF_Recv_Pkt:
Receiving MFF packet:
Src Interface : Ethernet1/3
Dst Interface :  N/A
InterfaceType :USER    Vlan ID :100
SrcMac :000d-5619-f7bc    SrcIp : 100.1.1.1
DstMac :0000-0000-0000    DstIp : 100.1.1.100
PacketType :REQUEST

// User port Ethernet1/3 received an ARP request from the host with IP address 100.1.1.1 and MAC
address 100.1.1.1 to the gateway at 100.1.1.100.
*Apr 26 12:12:59:523 2000 Sysname MFF/7/Debug_MFF_Send_Pkt:
Sending MFF packet:
Src Interface : N/A
Dst Interface : Ethernet1/2
InterfaceType : NETWORK Vlan ID : 100
SrcMac : 000d-5619-f7bc SrcIp : 100.1.1.1
DstMac : 0000-0000-0000 DstIp : 100.1.1.100
PacketType : REQUEST

// Network port Ethernet1/2 received an ARP request from the host with IP address 100.1.1.1 and MAC address 000d-5619-f7bc to the gateway at 100.1.1.100.

* Apr 26 12:12:59:534 2000 Sysname MFF/7/Debug_MFF_Recv_Pkt:
Receiving MFF packet:
Src Interface : Ethernet1/2
Dst Interface : N/A
InterfaceType : NETWORK Vlan ID : 100
SrcMac : 00e0-fc00-3102 SrcIp : 100.1.1.100
DstMac : 000d-5619-f7bc DstIp : 100.1.1.1
PacketType : REPLY

// Network port Ethernet1/2 received an ARP response from the gateway with IP address 100.1.1.100 and MAC address 00e0-fc00-3102 to the host with IP address 100.1.1.1 and MAC address 000d-5619-f7bc.

* Apr 26 12:12:59:553 2000 Sysname MFF/7/Debug_MFF_Send_Pkt:
Sending MFF packet:
Src Interface : Ethernet1/2
Dst Interface : N/A
InterfaceType : USER Vlan ID : 100
SrcMac : 00e0-fc00-3102 SrcIp : 100.1.1.100
DstMac : 0000-0000-0000 DstIp : 100.1.1.100

// Network port Ethernet1/2 received a gratuitous ARP packet from the gateway with IP address 100.1.1.100 and MAC address 00e0-fc00-3102.
MLD debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

**debugging mld**

Use **debugging mld** to enable MLD debugging.

Use **undo debugging mld** to disable MLD debugging.

**Syntax**

```plaintext
```

```plaintext
undo debugging mld { all | done | event | query [ receive | send ] | report | ssm-mapping | timer }
```

**Default**

MLD debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **all**: Specifies all types of MLD debugging.
- **done**: Specifies MLD done message debugging.
- **basic-acl6-number**: Specifies an IPv6 basic ACL by its number in the range of 2000 to 2999.
- **event**: Specifies MLD event debugging.
- **query**: Specifies MLD query message debugging.
- **advanced-acl6-number**: Specifies an IPv6 advanced ACL by its number, in the range of 3000 to 3999.
- **receive**: Specifies debugging for received MLD query messages.
- **send**: Specifies debugging for sent MLD query messages.
- **report**: Specifies MLD report debugging.
- **ssm-mapping**: Specifies MLD SSM mapping debugging.
- **timer**: Specifies MLD timer debugging.

**Usage guidelines**

Table 1 describes output fields and messages for the **debugging mld done** command.
Table 10 Output from the debugging mld done command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DONE</td>
<td>MLD done message.</td>
</tr>
<tr>
<td>Interfacename(ifadd)</td>
<td>Interface that received the MLD done message (interface address).</td>
</tr>
<tr>
<td>group gadd</td>
<td>IPv6 group address in the MLD done message.</td>
</tr>
<tr>
<td>Ignoring</td>
<td>This MLD done message was ignored.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the **debugging mld event** command.

Table 11 Output from the debugging mld event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Creating/creation/created aux join/aux prune adding/deleting downstream interface deleting/unregister/deleted Enqueue/Dequeueing Elected/ Un-elected | Event types:  
- Creating/creation/created  
- Aux join/aux prune  
- Adding/deleting downstream interface  
- Deleting/unregister/deleted  
- Enqueue/Dequeueing  
- Elected/ Un-elected |
| Interface interfacename(ifadd)     | Interface that responded to events (interface address). |
| (sadd, gadd)                       | (S, G) entry.                                     |
| (*, gadd)                          | (*, G) entry.                                     |
| Group limit reached on interface interfacename, failed to create group(gadd) | MLD failed to create a table entry for the group because the number of IPv6 multicast groups has reached the specified threshold. |

Table 3 describes output fields and messages for the **debugging mld query receive** command.

Table 12 Output from the debugging mld query receive command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version &lt;1-2&gt;</td>
<td>Version of the MLD query.</td>
</tr>
<tr>
<td>Interfacename(ifadd)</td>
<td>Interface that sent/received this message (interface address).</td>
</tr>
<tr>
<td>Ignoring</td>
<td>The MLD query message was ignored.</td>
</tr>
<tr>
<td>Received</td>
<td>The MLD query message was received.</td>
</tr>
<tr>
<td>Group gadd</td>
<td>IPv6 group address to be queried.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the **debugging mld report** command.

Table 13 Output from the debugging mld report command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignoring</td>
<td>This MLD report was ignored.</td>
</tr>
</tbody>
</table>
### IS_IN/IS_EX/TO_IN/TO_EX/ALLOW/BLOCK

Record types of MLDv2 membership reports:

- IS_IN
- IS_EX
- TO_IN
- TO_EX
- ALLOW
- BLOCK

### Group gadd

Reported IPv6 group address.

### (sadd, gadd)

(S, G) entry.

### v1/v2

Version of MLD membership report.

### Interfacename(ifadd)

Interface that sent/received this message (interface address).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version &lt;1-2&gt;</td>
<td>Version of the MLD query.</td>
</tr>
<tr>
<td>Interfacename(ifadd)</td>
<td>Interface that sent/received this message (interface address).</td>
</tr>
<tr>
<td>Ignoring</td>
<td>This MLD query message was ignored.</td>
</tr>
<tr>
<td>Send</td>
<td>The MLD query message was sent.</td>
</tr>
<tr>
<td>With/without s-bit</td>
<td>Whether S flag bit was set in the MLD query message.</td>
</tr>
<tr>
<td>Group gadd</td>
<td>IPv6 group address to be queried.</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the `debugging mld query send` command.

**Table 14 Output from the debugging mld query send command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssm-mapping notify</td>
<td>MLD reported SSM mapping multicast group information to the upper-layer routing protocol.</td>
</tr>
<tr>
<td>ssm-mapping group</td>
<td>SSM mapping multicast group.</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Group compatibility mode.</td>
</tr>
<tr>
<td>Lmqi timer</td>
<td>Last-listener query timer.</td>
</tr>
<tr>
<td>ssm-mapping policy table</td>
<td>SSM mapping rule table.</td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the `debugging mld ssm-mapping` command.

**Table 15 Output from the debugging mld ssm-mapping command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lmqi timeout for group</td>
<td>The last-listener query timer timed out.</td>
</tr>
</tbody>
</table>

Table 7 describes output fields and messages for the `debugging mld timer` command.

**Table 16 Output from the debugging mld timer command**
### Examples

# Enable MLD done message debugging. When MLD is enabled, output similar to the following example is generated:

```
<Sysname> debugging mld done
*0.167564 Sysname MLD/7/DONE:Received DONE for group(FF0E::101:101) on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00) (M053692)
// MLD received an MLD done message on Ethernet 1/1 with the group address FF0E::101:101.
```

# Enable MLD event debugging. When MLD is enabled, output similar to the following example is generated:

```
<Sysname> debugging mld event
*0.852518 Sysname MLD/7/EVENT:Elected querier on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00) (G10297)
// Ethernet 1/1 became the MLD querier.
```
Sysname MLD/7/EVENT: Enqueue group(FF0E::101:101) on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00) in group_calq. (G014076)

// MLD received an MLD done message on Ethernet 1/1 and reset the group aging timer.
*0.1224808 Sysname MLD/7/EVENT: Lmqi timeout for group(FF0E::101:101), sending last listener query on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00). (G013428)

// The last-listener query timer timed out. MLD sent out a multicast-address-specific query.
*0.1225820 Sysname MLD/7/EVENT: Group(FF0E::101:101) expired and sources empty. Deleting this group on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00). (G013318)

// No more join messages with the group address FF0E::101:101 were received on Ethernet 1/1. This entry timed out.
*0.1225820 Sysname MLD/7/EVENT: Deleting group(FF0E::101:101) on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00) (G014170)
*0.1225820 Sysname MLD/7/EVENT: Group(FF0E::101:101) deleted (G01805)

// IGMP deleted the entry for group FF0E::101:101.

# Enable debugging for received MLD query messages. When MLD is enabled, output similar to the following example is generated:
Sysname> debugging mld query receive

*0.2607914 Sysname MLD/7/QUERY: Received MLDv2 query message on Ethernet1/1(FE80::200:5EFF:FE01:6C00) from FE80::1 (M04430)

// MLD received an MLDv2 query on Ethernet 1/1.
*0.2607914 Sysname MLD/7/QUERY: Adopted querier's robustness(2), query interval(10) on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00) (M04984)

// MLD set the number of times to 2 and interval of sending multicast-address-specific queries to 10 seconds.

# Enable MLD membership report debugging. When MLD is enabled, output similar to the following example is generated:
Sysname> debugging mld report

*0.1988093 Sysname MLD/7/REPORT: Received IS_IN for group(FF1E::3) on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00) (M05704)

// MLD received an MLDv2 membership report for group FF1E::3 on Ethernet 1/1, with the record type of IS_IN.
*0.2103721 Sysname MLD/7/REPORT: Received IS_EX for group(FF1E::3) on interface Ethernet1/1(FE80::200:5EFF:FE01:6C00) (M05755)

// MLD received an MLDv2 membership report for group FF1E::3 on Ethernet 1/1, with the record type of IS_EX.

# Enable debugging for sent MLD query messages. When MLD is enabled, output similar to the following example is generated:
Sysname> debugging mld query send

*0.2237184 Sysname MLD/7/QUERY: Send version 2 general query on Ethernet1/1(FE80::200:5EFF:FE01:6C00) to destination(FF02::1) (M04791)

// MLD sent an MLDv2 general query message out of Ethernet 1/1

# Enable MLD SSM mapping debugging on the public network. When MLD SSM mapping is enabled, output similar to the following example is generated:
Sysname> debugging mld ssm-mapping
Sysname MLD/7/SSM-MAPPING:(public net):Receive ssm-mapping report for group FF34::1 on interface Ethernet1/1 (FE80::200:5EFF:FE01:6C00) (M053895)
// MLD received an MLDv1 membership report with the multicast group address FF34::1 on Ethernet 1/1. The multicast group address is in the MLD SSM mapping rule range.

Sysname MLD/7/SSM-MAPPING:(public net):Creating ssm-mapping group (FF34::1) for interface Ethernet1/1 (FE80::200:5EFF:FE01:6C00) (G015470)
// MLD created an SSM mapping multicast group FF34::1 on Ethernet 1/1 after source addresses were mapped with the MLD SSM mapping rule.

Sysname MLD/7/SSM-MAPPING:(public net):Receive ssm-mapping LEAVE for group FF34::1 on interface Ethernet1/1 (FE80::200:5EFF:FE01:6C00) (M054112)
// MLD received an MLDv1 done message with the multicast group address of FF34::1 on Ethernet 1/1. The multicast group address is in the MLD SSM mapping rule range.

Sysname MLD/7/SSM-MAPPING:(public net):Ssm-mapping LEAVE received for group FF34::1 is immediately processed on interface Ethernet1/1 (FE80::200:5EFF:FE01:6C00) (M054185)
// The MLD SSM snooping multicast group was directly deleted because fast-leave processing has been enabled.

Sysname MLD/7/SSM-MAPPING:(public net):Deleting ssm-mapping group (FF34::1) on interface Ethernet1/1 (FE80::200:5EFF:FE01:6C00) (G015835)
// MLD deleted the MLD SSM mapping multicast group on Ethernet 1/1.

# Enable MLD timer debugging. When MLD is enabled, output similar to the following example is generated:
<Sysname> debugging mld timer

*0.2955218 Sysname MLD/7/TIMER:Deleting v1host timer for group (FF0E::101:101) (M053388)
// The MLDv1 host timer timed out.

*0.3023658 Sysname MLD/7/TIMER:Deleting source lmqi timer for group (FF0E::101:101) on interface Ethernet1/1 (FE80::200:5EFF:FE01:6C00) (M053493)
// The last-listener query timer timed out.

*0.40447221 Sysname MLD/7/TIMER:(public net):Setting source expiry timer for source FE90::200:5EFF:FE01:6C00 of group FF34::1 on Ethernet1/1 (FE80::6C00) to 260 (M054026)
// MLD received an MLDv1 membership report with the multicast group address FF34::1 on Ethernet 1/1. The multicast group address is in the MLD SSM mapping rule range. After the MLD SSM mapping multicast group was created, an aging timer was set for the group per IPv6 multicast source.

*0.40447221 Sysname MLD/7/TIMER:(public net):Lmqi timeout for group FF34::1, sending last member query on Ethernet1/1 (FE80::6C00) (G016089)
// MLD received an MLDv1 done message with the multicast group address of FF34::1 on Ethernet 1/1. The multicast group address is in the MLD SSM mapping rule range. After the last-listener query timer of the MLD SSM mapping multicast group timed out, a multicast-address-specific query was sent.

*0.70447221 Sysname MLD/7/TIMER:(public net):Deleting lmqi timer for ssm-mapping group FF34::1 on Ethernet1/1 (FE80::6C00) (G016110)
// MLD sent out queries. No report was received, so the last-listener query timer was deleted.

debugging mld proxying

Use debugging mld proxying to enable MLD proxy debugging.

Use undo debugging mld proxying to disable MLD proxy debugging.
Syntax

```
debugging mld proxying { event | packet | routing-table } advanced-acl6-number
undo debugging mld proxying { event | packet | routing-table }
```

Default

MLD proxy debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- **event**: Specifies MLD proxy event debugging.
- **packet**: Specifies MLD proxy packet debugging.
- **routing table**: Specifies MLD proxy routing table debugging.
- **advanced-acl6-number**: Specifies an IPv6 advanced ACL by its number, in the range of 3000 to 3999.

Usage guidelines

Table 8 describes output fields and messages for the `debugging mld proxying event` command.

**Table 17 Output from the debugging mld proxying event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enqueued</td>
<td>The multicast group was enqueued.</td>
</tr>
<tr>
<td>dequeued</td>
<td>The multicast group was dequeued.</td>
</tr>
<tr>
<td>proxy_calq</td>
<td>MLD proxy CALQ queue.</td>
</tr>
<tr>
<td>proxy cache</td>
<td>MLD proxy cache.</td>
</tr>
<tr>
<td>proxy database</td>
<td>Group membership database maintained by the MLD proxy device.</td>
</tr>
<tr>
<td>Number of joins</td>
<td>Number of joined sources.</td>
</tr>
</tbody>
</table>

Table 9 describes output fields and messages for the `debugging mld proxying packet` command.

**Table 18 Output from the debugging mld proxying packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxying interface</td>
<td>MLD proxy interface.</td>
</tr>
<tr>
<td>query</td>
<td>MLD query message.</td>
</tr>
<tr>
<td>report</td>
<td>MLD report message.</td>
</tr>
<tr>
<td>record</td>
<td>Group record.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>type 1</td>
<td>Group record types:</td>
</tr>
<tr>
<td>type 2</td>
<td>• Type 1—IS_IN.</td>
</tr>
<tr>
<td>type 3</td>
<td>• Type 2—IS_EX.</td>
</tr>
<tr>
<td>type 4</td>
<td>• Type 3—TO_IN.</td>
</tr>
<tr>
<td>type 5</td>
<td>• Type 4—TO_EX.</td>
</tr>
<tr>
<td>type 6</td>
<td>• Type 5—ALLOW.</td>
</tr>
<tr>
<td>type 7</td>
<td>• Type 6—BLOCK.</td>
</tr>
<tr>
<td>type 8</td>
<td>• Type 7—JOIN.</td>
</tr>
<tr>
<td></td>
<td>• Type 8—LEAVE.</td>
</tr>
</tbody>
</table>

Table 10 describes output fields and messages for the `debugging mld proxying routing-table` command.

**Table 19 Output from the debugging mld proxying routing-table command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(sadd, gad)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>Recovery of route</td>
<td>The route was recovered.</td>
</tr>
<tr>
<td>Loss of route</td>
<td>The route was lost.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable MLD proxy event debugging. When MLD proxy is enabled on the interface, output similar to the following example is generated:

```
<Sysname> debugging mld proxying event
*0.1631452 Sysname MLD/7/PRY_EVT:(public net): Group FF0E::101:101 added in proxy cache. (G363219)
// MLD received a join message for group FF0E::101:101 on a downstream interface and added this multicast group into the MLD proxy cache.
*0.1631831 Sysname MLD/7/PRY_EVT:(public net): EXCLUDE group FF0E::101:101 added in proxy database.
// MLD added an entry for the multicast group in the EXCLUDE mode to the group membership database in the MLD proxy device.
*0.1632021 Sysname MLD/7/PRY_EVT:(public net): Number of joins increased to 2 for group FF0E::101:101 in proxy database.
// MLD received a join message for the multicast group FF0E::101:101 on another downstream interface. The number of records for the same group in the database increased to two.
*0.1678546 Sysname MLD/7/PRY_EVT: Group FF0E::101:101 enqueued in proxy_calq. (M363784)
// Upon receiving a query message for group FF0E::101:101, the upstream interface started the delay timer and put this group in the CALQ queue.
*0.1714536 Sysname MLD/7/PRY_EVT: Group FF0E::101:101 dequeued from proxy_calq. (G363833)
// When the upstream interface went down, it removed the group from the CALQ queue.
*0.5532846 Sysname MLD/7/PRY_EVT:(public net): Group record created for group FF0E::101:101 at delay timer timeout.
// The delay timer timed out, and a group record was created for the multicast group FF0E::101:101.
```
# Enable MLD proxy packet debugging on the public network. When MLD proxy is enabled on the interface, output similar to the following example is generated:

```
<Sysname> debugging mld proxying packet
*0.1726142 Sysname MLD/7/PRY_EVT: Received MLDv1 query for group FF0E::101:101 on version 1 proxying interface Ethernet1/1(FE80::200:6C00). (G362885)

// MLD received an MLDv1 query message for group FF0E::101:101 on the upstream interface Ethernet1/1.
*0.1754567 Sysname MLD/7/PRY_EVT: MLDv1 report for group FF0E::101:101 sent out Ethernet1/1(FE80::200:6C00). (G363449)

// After the entry for group FF0E::101:101 was added, the proxy device sent an MLD report for group FF0E::101:101 on Ethernet 1/1.
*0.1752533 Sysname MLD/7/PRY_EVT: Type 3 record added for group FF0E::101:101. (G363564)

// When an MLD report was sent as a result of a change in the membership for the multicast group FF0E::101:101, a TO-IN group record for the group was added.
*0.1783136 Sysname MLD/7/PRY_EVT: Source 100::10 added to group record. (G363647)

// The source 100::10 was added into the group record.
*0.1790527 Sysname MLD/7/PRY_EVT: MLDv2 report sent. (G363662)

// When the membership for the group changed, an MLDv2 report was sent to the upstream device.
```

# Enable MLD proxy route debugging. When MLD proxy is enabled on the interface, output similar to the following example is generated:

```
<Sysname> debugging mld proxying routing-table
*0.57932162 Sysname MLD/7/PRY_RT:(public net): Entry (100::1, FF3E::1) created in IPv6 multicast routing table. (G01597)

// MLD received an (S, G) join with the group address in the SSM group range on a downstream interface and created a forwarding entry.
*0.58934175 Sysname MLD/7/PRY_RT:(public net): Entries for group FF3E::1 removed upon change of SSM group range. (G36139)

// MLD detected a change in the SSM group policy and removed entries for the group FF3E::1.
*0.58034187 Sysname MLD/7/PRY_RT:(public net): Loss of route for (100::1, FF3E::1) detected, (S,G) entry removed. (G361460)

// MLD detected that the route was lost and removed the associated forwarding entry.
*0.58034165 Sysname MLD/7/PRY_RT:(public net): Recovery of route for (100::1, FF3E::1) detected, (S,G) entry created. (G361460)

// MLD detected that the route was recovered and created the associated forwarding entry.
```
MLD snooping debugging commands

The MCS module name is identified as “multicast” in debugging messages.

The output description tables in this document only contain fields and messages that require an explanation.

Some information in this chapter is device type-specific. Devices in this chapter are categorized depending on their IRF capability and support for interface cards that use independent processors for forwarding traffic, as shown in Table 1.

**Table 20 Device types**

<table>
<thead>
<tr>
<th>Device type</th>
<th>Interface cards with on-card processors</th>
<th>IRF capability</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed devices</td>
<td>Yes</td>
<td>No</td>
<td>HP 6600 routers (except for 6602)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes (in standalone mode)</td>
<td>HP 12500 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 10500 switches</td>
</tr>
<tr>
<td>Distributed IRF devices</td>
<td>Yes</td>
<td>Yes (in IRF mode)</td>
<td>HP 12500 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 10500 switches</td>
</tr>
<tr>
<td>Centralized devices</td>
<td>No</td>
<td>No</td>
<td>HP MSR routers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 6602 router</td>
</tr>
<tr>
<td>Centralized IRF devices</td>
<td>No</td>
<td>Yes</td>
<td>HP 5800 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 5500 switches</td>
</tr>
</tbody>
</table>

**debugging mld-snooping**

Use **debugging mld-snooping** to enable MLD snooping debugging.

Use **undo debugging mld-snooping** to disable MLD snooping debugging.

**Syntax**

Centralized devices:

```
debugging mld-snooping { abnormal | all | driver | event | group | packet [ vlan vlan-id ] [ port interface-type interface-number ] | timer }
undo debugging mld-snooping { abnormal | all | driver | event | group | packet | timer }
```

Distributed devices/centralized IRF devices/distributed IRF devices:

```
debugging mld-snooping { abnormal | all | driver | event | group | ipc { receive | send } | packet [ vlan vlan-id ] [ port interface-type interface-number ] | timer }
undo debugging mld-snooping { abnormal | all | driver | event | group | ipc { receive | send } | packet | timer }
```

**Default**

MLD snooping debugging is disabled.
Views

User view

Default command level

1: Monitor level

Parameters

 abnormal: Specifies MLD snooping exception debugging.
 all: Specifies all types of MLD snooping debugging.
 driver: Specifies MLD snooping interface driver debugging.
 event: Specifies MLD snooping event debugging.
 group: Specifies MLD snooping multicast group debugging.
 ipc { receive | send }: Specifies inbound/outbound IPC packet debugging.
 packet: Specifies MLD snooping packet debugging.
 vlan vlan-id: Specifies the VLAN to which an Ethernet port belongs. The value range for the vlan-id argument is 1 to 4094.
 port interface-type interface-number: Specifies an Ethernet port.
 timer: Specifies debugging for MLD snooping timers.

Usage guidelines

Table 2 describes output fields and messages for the debugging mld-snooping driver command.

Table 21 Output from the debugging mld-snooping driver command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add/Delete multicast vlan OIF to/from driver</td>
<td>A multicast VLAN outgoing interface was added to or deleted from the driver.</td>
</tr>
<tr>
<td>Add/Delete Level2 IP entry to/from driver</td>
<td>A Layer 2 IP entry was added to or deleted from the driver.</td>
</tr>
<tr>
<td>Add/Delete Unit/Board for multicast vlan to/from driver</td>
<td>A unit or board for the multicast VLAN was added to or deleted from the driver.</td>
</tr>
<tr>
<td>Add/Delete Unit/Board for Level2 IP entry to/from driver</td>
<td>A unit or board for the Layer 2 IP entry was added to or removed from the driver.</td>
</tr>
<tr>
<td>Add/Delete Port for multicast vlan OIF to/from driver</td>
<td>A port was added to or deleted from the multicast VLAN outgoing interface list on the driver.</td>
</tr>
<tr>
<td>Add/Delete Port for Level2 IP entry to/from driver</td>
<td>A port was added to or deleted from the Layer 2 IP entry on the driver.</td>
</tr>
<tr>
<td>Add/Delete multicast vlan IP entry to/from driver</td>
<td>A multicast VLAN IP entry was added to or deleted from the driver.</td>
</tr>
<tr>
<td>Add/Delete Level2 MAC entry to/from driver</td>
<td>A Layer 2 MAC entry was added to or deleted from the driver.</td>
</tr>
<tr>
<td>mask boardmask</td>
<td>The destination board mask of an entry was added to or deleted from the driver.</td>
</tr>
<tr>
<td>(saddr, gaddr)</td>
<td>(S, G) entry.</td>
</tr>
</tbody>
</table>
Table 3 describes output fields and messages for the **debugging mld-snooping event** command.

### Table 22 Output from the debugging mld-snooping event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Succeed to enable/disable MLD snooping</td>
<td>MLD snooping was globally enabled or disabled.</td>
</tr>
<tr>
<td>globally</td>
<td></td>
</tr>
<tr>
<td>Succeed to enable/disable MLD snooping</td>
<td>The function of dropping unknown multicast packets was globally enabled or</td>
</tr>
<tr>
<td>drop-unknown globally</td>
<td>disabled.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Succeed to enable/disable MLD snooping</td>
<td>MLD snooping proxy was enabled or disabled in a VLAN.</td>
</tr>
<tr>
<td>proxying on vlan vlanid</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Received chassis/slot/interface event</td>
<td>An IRF member device, card, or VLAN interface event was received.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the **debugging mld-snooping group** command.

### Table 23 Output from the debugging mld-snooping group command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create L2 IP Source entry</td>
<td>MLD snooping created a Layer 2 IP entry.</td>
</tr>
<tr>
<td>Deleted IP and MAC entry</td>
<td>MLD snooping deleted the IP and MAC entries.</td>
</tr>
<tr>
<td>(saddr, gaddr) on vlan vlanid</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td></td>
<td>VLAN to which the (S, G) belongs.</td>
</tr>
<tr>
<td>Group number exceeds the max number</td>
<td>The number of multicast groups exceeded the maximum number.</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the **debugging mld-snooping ipc** command.

### Table 24 Output from the debugging mld-snooping ipc command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inform other boards/units: event</td>
<td>MLD snooping informed the other boards/units about the event.</td>
</tr>
<tr>
<td>Successful to send ipc information to</td>
<td>MLD snooping sent IPC packets to the IPC queue successfully.</td>
</tr>
<tr>
<td>ipc queue</td>
<td></td>
</tr>
<tr>
<td>Received an IPC packet and insert it into</td>
<td>MLD snooping received an IPC packet and successfully inserted it into the</td>
</tr>
<tr>
<td>queue</td>
<td>IPC queue.</td>
</tr>
<tr>
<td>Received an IPC packet, but failed to</td>
<td>MLD snooping received an IPC packet but failed to insert it into the IPC</td>
</tr>
<tr>
<td>insert it into queue</td>
<td>queue.</td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the **debugging mld-snooping packet** command.

### Table 25 Output from the debugging mld-snooping packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port portname</td>
<td>Port that processed the message.</td>
</tr>
<tr>
<td>on vlan vlanid</td>
<td>ID of the VLAN to which the port belongs.</td>
</tr>
<tr>
<td>source address addr</td>
<td>Source address of the packet.</td>
</tr>
<tr>
<td>destination address addr</td>
<td>Destination address of the message.</td>
</tr>
</tbody>
</table>
Table 7 describes output fields and messages for the **debugging mld-snooping timer** command.

### Table 26 Output from the debugging mld-snooping timer command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create/Set</td>
<td>MLD snooping created or set a timer.</td>
</tr>
<tr>
<td>host port aging timer/time</td>
<td>MLD snooping timers:</td>
</tr>
<tr>
<td>router port aging timer/time</td>
<td>• Host port aging timer/time</td>
</tr>
<tr>
<td>aggregation host port aging</td>
<td>• Router port aging timer/time</td>
</tr>
<tr>
<td>aggregation router port</td>
<td>• Aggregation host port aging timer/time</td>
</tr>
<tr>
<td>querier timer/interval</td>
<td>• Aggregation router port aging timer/time</td>
</tr>
<tr>
<td>query respond timer</td>
<td>• Querier timer/interval</td>
</tr>
<tr>
<td></td>
<td>• Query respond timer</td>
</tr>
</tbody>
</table>

**Examples**

# Enable MLD snooping interface driver debugging. When MLD snooping is enabled on the device, output similar to the following example is generated:

```
<Sysname> debugging mld-snooping driver
*Nov 7 09:32:30:738 2006 Sysname MCS/7/driver:Add Level2 IP entry to driver: vlan:2, S,G(2002::2,FF08::8), port num:1. (G08758)

// An IP entry (2002::2, FF08::8) was added to the driver in VLAN 2.
```

# Enable MLD snooping event debugging. When MLD snooping is on the device and MLD snooping proxy is enabled in VLAN 2, output similar to the following example is generated:

```
<Sysname> debugging mld-snooping event
*Nov 7 17:22:24:532 2006 Sysname MCS/7/event:Succeed to enable MLD snooping globally. (G083769)

// MLD snooping was enabled globally.
```

*Nov 7 17:23:21:742 2006 Sysname MCS/7/event:Succeed to enable MLD snooping proxying on vlan 2. (G202463)

// MLD snooping proxy was enabled in VLAN 2.

# Enable MLD snooping multicast group debugging. When MLD snooping is enabled on the device, output similar to the following example is generated:

```
<Sysname> debugging mld-snooping group
*Nov 7 09:34:59:488 2006 Sysname MCS/7/groups:Deleted IP and MAC entry of (S,G) on vlan 2. (G142433)

// The IP and MAC entry was deleted.
```

# Enable inbound IPC packet debugging. When MLD snooping is enabled on the device, output similar to the following example is generated:

```
<Sysname> debugging mld-snooping ipc receive
*Nov 7 18:07:02:726 2006 Sysname MCS/7/ipcrecv:Slot=1;Received an IPC packet and insert it into queue. (G18955)

// MLD snooping received an IPC packet and inserted it into the IPC queue.
```

# Enable outbound IPC packet debugging. When MLD snooping is enabled on the device, output similar to the following example is generated:

```
<Sysname> debugging mld-snooping ipc send
```
**Nov 7 17:06:08:412 2006 Sysname MCS/7/ipcsend:** Inform other boards: this board is a new router board. (G143331)

// MLD snooping sent an IPC packet.

# Enable MLD snooping packet debugging. When MLD snooping is enabled on the device, output similar to the following example is generated:

<Sysname> debugging mld-snooping packet

**Nov 7 09:37:21:174 2006 Sysname MCS/7/packet:** Received MLDv2 report packet for destination address FF02::16 from port Eth1/1 on vlan 2. (G213068)

// MLD snooping received an MLD report on Ethernet 1/1 in VLAN 2.

**Nov 7 09:37:54:174 2006 Sysname MCS/7/packet:** Send the packet to port Eth1/1 of vlan 2, which source address is FE80::1 and destination address is FF02::16. (G261104)

// MLD snooping sent an MLD message with the source address FE80::1 and the destination address FF02::16 out of Ethernet 1/1 in VLAN 2.

# Enable MLD snooping timer debugging. When MLD snooping is enabled on the device, output similar to the following example is generated:

<Sysname> debugging mld-snooping timer

**Nov 7 17:06:08:412 2006 Sysname MCS/7/timer:** Create router port aging timer, 260 seconds, for port: Eth1/1, vlan: 2. (G141414)

// MLD snooping created a router port aging timer on Ethernet 1/1 in VLAN 2 and set the aging timer to 260 seconds.

**Nov 7 17:06:54:712 2006 Sysname MCS/7/timer:** Create query respond timer of group FF02::16, 2000 milliseconds. (G153777)

// MLD snooping created a query response timer for the IPv6 multicast group FF02::16 and set the time interval to 2 seconds.
Modem debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging modem

Use debugging modem to enable modem debugging.

Use undo debugging modem to disable modem debugging.

Syntax

debugging modem [ interface interface-type interface-number ]
undo debugging modem [ interface interface-type interface-number ]

Views

User view

Default command level

1: Monitor level

Parameters

interface-type interface-number: Interface type and interface number.

Usage guidelines

Table 1 describes output fields and messages for the debugging modem command.

Table 27 Output from the debugging modem command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>AT command.</td>
</tr>
<tr>
<td>OK</td>
<td>Connection was established successfully.</td>
</tr>
<tr>
<td>ATDT</td>
<td>AT command used to dial a number.</td>
</tr>
<tr>
<td>ATA</td>
<td>AT command used to answer a call without waiting for a ring.</td>
</tr>
<tr>
<td>+++</td>
<td>Quit command.</td>
</tr>
<tr>
<td>IDLE</td>
<td>Idle state.</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Connected state.</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>Active state.</td>
</tr>
<tr>
<td>DISCONNECT</td>
<td>Disconnect state.</td>
</tr>
<tr>
<td>BUSY</td>
<td>Line busy.</td>
</tr>
<tr>
<td>NO CARRIER</td>
<td>No carrier signal was detected.</td>
</tr>
<tr>
<td>MDM_LINE_MODEM_NO</td>
<td>Both answering calls and placing calls were disabled.</td>
</tr>
<tr>
<td>MDM_LINE_MODEM_IN</td>
<td>Answering calls was enabled.</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
MDM_LINE_MODEM_OUT | Placing calls was enabled.
MDM_LINE_MODEM_INOUT | Both answering calls and placing calls were enabled.
MDM_LINE_DISCONNECT | Modem was disconnected.
MDM_LINE_CALLBACK | Modem callback.
MDM_DDR_CONN_REQ | DDR connection request was received.
MDM_DDR_CONN_RES | DDR connection acknowledgement was received.
MDM_DDR_DISC_REQ | DDR disconnect request was received.
MDM_DDR_DISC_RES | DDR disconnect acknowledgement was received.
MDM_PHY_SHUT | SHUT command was issued.
MDM_PHY_NO_SHUT | No SHUT command was issued.
MDM_PHY_TO_AT | Modem changed to the AT command mode.
MDM_PHY_CD_DOWN | CD signal went invalid.
MDM_PHY_CD_UP | CD signal went valid.

### Examples

Output in the following example was created under the following conditions:

- Router A and Router B are connected through a PBX.
- Router A has the following configuration:
  ```
  #
  dialer-rule 1 ip permit
  #
  interface Analogmodem5/1
  async mode protocol
  link-protocol ppp
  ip address 51.0.0.1 255.255.255.0
  dialer enable-circular
  dialer-group 1
  dialer number 105
  #
  user-interface tty 177
  modem both
  #
  ```
- Router B has the following configuration:
  ```
  #
  dialer-rule 1 ip permit
  #
  interface Analogmodem5/1
  async mode protocol
  link-protocol ppp
  ip address 51.0.0.2 255.255.255.0
  dialer enable-circular
  dialer-group 1
  ```
dialer number 106
#
user-interface tty 177
modem both
#

# Enable modem debugging on Router A, and then ping Router B from Router A.
<Sysname> debugging modem interface Analogmodem 5/1
<Sysname> ping -c 1 51.0.0.2
*Mar 31 12:13:00:352 2006 Sysname MODEM/7/debug_CTRL:
  Analogmodem5/1 Modem received information: MDM_DDR_CONN_REQ
  // Analogmodem 5/1 received a connection request.
*Mar 31 12:13:00:352 2006 Sysname MODEM/7/debug_AT_Send:
  Analogmodem5/1 Data of AT is sent: ATDT105
  // Analogmodem 5/1 sent the number to be called.
*Mar 31 12:13:00:352 2006 Sysname MODEM/7/debug_ModemState:
  Analogmodem5/1 The modem state is changed: from IDLE to CONNECT
  // Analogmodem 5/1 turned from idle state to connected state.
*Mar 31 12:13:00:379 2006 Sysname MODEM/7/debug_AT_Recv:
  Analogmodem5/1 Data of AT is received: CONNECT 57600
  // Analogmodem 5/1 received an AT command: connect 57600.
*Mar 31 12:13:25:780 2006 Sysname MODEM/7/debug_CTRL:
  Analogmodem5/1 Modem received information: MDM_PHY_CD_UP
  // Analogmodem 5/1 received a message that the CD signal was valid.
*Mar 31 12:13:25:886 2006 Sysname MODEM/7/debug_ModemState:
  Analogmodem5/1 The modem state is changed: from CONNECT to ACTIVE
  // Analogmodem 5/1 changed from connected state to active state.
*Mar 31 12:13:33:630 2006 Sysname MODEM/7/debug_CTRL:
  Analogmodem5/1 Modem received information: MDM_DDR_DISC_REQ
  // Analogmodem 5/1 received a disconnect request. This message is displayed when the dialer disconnect command is executed to disconnect the modem.
*Mar 31 12:13:33:738 2006 Sysname MODEM/7/debug_ModemState:
  Analogmodem5/1 The modem state is changed: from ACTIVE to DISCONNECT
  // Analogmodem 5/1 changed from active state to disconnect state.
*Mar 31 12:13:33:738 2006 Sysname MODEM/7/debug_AT_Send:
  Analogmodem5/1 Data of AT is sent: +++
  // Analogmodem 5/1 issued a command to terminate the connection.
*Mar 31 12:13:34:583 2006 Sysname MODEM/7/debug_AT_Recv:
  Analogmodem5/1 Data of AT is received: NO CARRIER
  // Analogmodem 5/1 received a no-carrier AT data command.
*Mar 31 12:13:35:750 2006 Sysname MODEM/7/debug_ModemState:
  Analogmodem5/1 The modem state is changed: from DISCONNECT to IDLE
// Analogmodem 5/1 changed from disconnect state to idle state.
*Mar 31 12:13:35:750 2006 Sysname MODEM/7/debug_ModemState:
  Analogmodem5/1  The modem state is changed: from IDLE to DISCONNECT

// Analogmodem 5/1 changed from idle state to disconnect state.
*Mar 31 12:13:35:750 2006 Sysname MODEM/7/debug_AT_Send:
  Analogmodem5/1  Data of AT is sent: +++

// Analogmodem 5/1 issued a command to terminate the connection.
*Mar 31 12:13:35:751 2006 Sysname MODEM/7/debug_ModemState:
  Analogmodem5/1  The modem state is changed: from DISCONNECT to IDLE

// Analogmodem 5/1 changed from disconnect state to idle state.
Monitor Link debugging commands

The Monitor Link module name is identified as “MTLK” in debugging messages.

debugging monitor-link

Use `debugging monitor-link` to enable Monitor Link debugging.

Use `undo debugging monitor-link` to disable Monitor Link debugging.

Syntax

```
debugging monitor-link [ group group-id ] { all | error | event }
undo debugging monitor-link [ group group-id ] { all | error | event }
```

Default

Monitor link debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `group group-id`: Specifies a monitor link group by its ID. If no group ID is specified, this command enables or disables monitor link debugging for all monitor link groups.
- `all`: Specifies all types of monitor link group debugging.
- `error`: Specifies monitor link group error debugging.
- `event`: Specifies monitor link group event debugging.

Examples

```
# Enable event debugging for monitor link group 1.
<Sysname> debugging monitor-link group 1 event
*Dec 28 19:37:47:543 2007 Sysname MTLK/7/GROUPSTATE:
Monitor link group 1 is down
// Monitor link group 1 went down.
```
MPLS L2VPN debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debbuging mpls l2vpn

Use `debugging mpls l2vpn` to enable debugging for MPLS L2VPN.
Use `undo debugging mpls l2vpn` to disable debugging for MPLS L2VPN.

Syntax

```
debugging mpls l2vpn { advertisement | all | connections | error | event | hsb }
undo debugging mpls l2vpn { advertisement | all | connections | error | event | hsb }
```

Default

Debugging for MPLS L2VPN is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- **advertisement**: Specifies debugging for advertisement events.
- **all**: Specifies debugging for all events.
- **connections**: Specifies debugging for connection events.
- **error**: Specifies debugging for error events.
- **event**: Specifies debugging for bottom layer events.
- **hsb**: Specifies debugging for hot standby events.

Usage guidelines

Table 1 describes output fields and messages for the `debugging mpls l2vpn advertisement` command.

**Table 28 Output from the debugging mpls l2vpn advertisement command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>peer pe ip address</td>
<td>IP address of the specified peer PE.</td>
</tr>
<tr>
<td>fec element type</td>
<td>FEC type. The value is 128.</td>
</tr>
<tr>
<td>c-Bit</td>
<td>Control bit. 1 means enabled, and 0 means not enabled.</td>
</tr>
<tr>
<td>group id</td>
<td>Used to remove VC information with the same group ID in batch.</td>
</tr>
<tr>
<td>vsi type</td>
<td>VSI access type.</td>
</tr>
</tbody>
</table>
Table 29 Output from the debugging mpls l2vpn event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_ulL2vpnMsg2MFWCounter</td>
<td>L2VPN advertised counter to MFW.</td>
</tr>
<tr>
<td>g_ulL2vpnMsg2BGPCounter</td>
<td>L2VPN advertised counter to BGP.</td>
</tr>
<tr>
<td>g_ulL2vpnMsg2LDPCounter</td>
<td>L2VPN advertised counter to LDP.</td>
</tr>
<tr>
<td>intf Up event receive&amp;process</td>
<td>L2VPN received and processed interface up event.</td>
</tr>
<tr>
<td>intf Notify Msg to IFNET process</td>
<td>Interface notified IFNET process of a message.</td>
</tr>
</tbody>
</table>

Table 30 Output from the debugging mpls l2vpn hsb command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2VPN CCCEncapInf RealTime Backup Modify IfState....entered</td>
<td>L2VPN interface encapsulation state change realtime backup began.</td>
</tr>
<tr>
<td>L2VPN CCCEncapInf RealTime Backup Modify IfState....leaving</td>
<td>L2VPN interface encapsulation state change realtime backup finished.</td>
</tr>
<tr>
<td>L2VPN Localldp Realtime backup Modify VCState....entered</td>
<td>L2VPN local LDP information realtime backup began.</td>
</tr>
<tr>
<td>L2VPN Localldp Realtime backup Modify VCState....leaving</td>
<td>L2VPN local LDP information realtime backup finished.</td>
</tr>
<tr>
<td>L2VPN Restore Scheduler....entered</td>
<td>L2VPN restore began.</td>
</tr>
<tr>
<td>L2VPN Restore Buffer....entered</td>
<td>L2VPN restore buffer began.</td>
</tr>
<tr>
<td>L2VPN Parse Restore Buffer....entered</td>
<td>L2VPN parse restore buffer began.</td>
</tr>
<tr>
<td>L2VPN CCCEncapInfDLL Restore....entered</td>
<td>L2VPN interface encapsulation restore began.</td>
</tr>
<tr>
<td>L2VPN CCCEncapInfDLL Restore....leaving</td>
<td>L2VPN interface encapsulation restore finished.</td>
</tr>
<tr>
<td>L2VPN Parse Restore Buffer....leaving</td>
<td>L2VPN parse restore buffer finished.</td>
</tr>
<tr>
<td>L2VPN Restore Buffer....leaving</td>
<td>L2VPN restore buffer finished.</td>
</tr>
<tr>
<td>L2VPN Restore Scheduler....leaving</td>
<td>L2VPN restore began.</td>
</tr>
<tr>
<td>L2VPN BGPVC Batch Backup....entered</td>
<td>L2VPN BGP VC batch backup began.</td>
</tr>
<tr>
<td>L2VPN BGPVC Batch Backup....leaving</td>
<td>L2VPN BGP VC batch backup finished.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the debugging mpls l2vpn connection command.
Table 31 Output from the debugging mpls l2vpn connection command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RmtCEID</td>
<td>Remote CE identifier.</td>
</tr>
<tr>
<td>vcid</td>
<td>VC identifier.</td>
</tr>
<tr>
<td>vctype</td>
<td>VC type.</td>
</tr>
<tr>
<td>dest</td>
<td>Destination LSR address.</td>
</tr>
</tbody>
</table>

Table 32 Output from the debugging mpls l2vpn error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Invalid Remote interface - not able to get the outifindex</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the debugging mpls l2vpn error command.

Examples

# Enable debugging for MPLS L2VPN advertisement. Output similar to the following example is generated when these tasks are performed:

- Connect three routers that are directly connected.
- Configure MPLS, MPLS LDP, and Martini settings on each router, and establish a Martini connection between two routers.
- Execute the debugging mpls l2vpn advertisement command on one PE router.
- Shut down the Martini-enabled interface that is connected to the private network.

```bash
<PE1> debugging mpls l2vpn advertisement
<PE1> terminal monitor
<PE1> terminal debugging
<PE1> system-view
[PE1] interface vlan-interface 80
[PE1-Vlan-interface80] display this
#
interface Vlan-interface80
  ip address 100.5.1.2 255.255.255.0
  mpls l2vc 33.33.33.33 100
#
return
[PE1-Vlan-interface80]
[PE1-Vlan-interface80]
[PE1-Vlan-interface80] shutdown
[PE1-Vlan-interface80]
Jul  6 17:56:30:172 2006 PE1 IFNET/5/LINK UPDOWN:
  Vlan-interface80: link status is DOWN
*0.3799203 PE1 L2V/8/DBG:
  ! [L2VPN-LDP]Info : sent label withdraw message. context of message as follow: 
```

43
peer pe ip address is: 33.33.33.33
request_id is: 0
label is: 1024
status_code is: 0x0
fec element type is: 128
c-Bit is: 0
vc type is: ethernet
vc info length is: 4
group id is: 0
vc id is: 100
length of vc_fec_tlv is: 12
context of vc_fec_tlv is: 80 00 05 04 00 00 00 00 00 00 00 64

*0.3799203 PE1 L2V/8(DBG):
! L2VPN LDP - Send LABEL WITHDRAW to ldp success

// L2VPN sent a label withdraw message to LDP successfully.
*0.3799375 PE1 L2V/8(DBG):
! [L2VPN-LDP]Info :
received label release message.
context of message as follow:
-------------------------------
peer pe ip address is: 33.33.33.33
request_id is: 0
label is: 1024
status_code is: 0x0
fec element type is: 128
c-Bit is: 0
vc type is: ethernet
vc info length is: 4
group id is: 0
vc id is: 100
length of vc_fec_tlv is: 12
context of vc_fec_tlv is: 80 00 05 04 00 00 00 00 00 00 00 64

// L2VPN received a label release message.
*0.3799375 PE1 L2V/8(DBG):

! [L2VPN-VSI]Info :
VSI: received label release message.
context of message as follow:
-------------------------------
peer pe ip address is: 33.33.33.33
request_id is: 0
label is: 1024
status_code is: 0x0
fec element type is: 128
c-Bit is: 0
vsi type is: ethernet
vsi info length is: 4
group id is:                  0
vsi id is:                    100
length of vc_fec_tlv is:      12
context of vc_fec_tlv is:     80 00 05 04 00 00 00 00 00 00 00 64

// The VSI at the private network side received the label release message.
*0.3799422 PE1 L2V/8/DBG:
  ! L2VPN LDP no corresponding local vc

// L2VPN LDP had no corresponding local VC.
*0.3799422 PE1 L2V/8/DBG:
  ! L2VPN VSI no local ldp peer found.

// L2VPN VSI had no local LDP peer found.

# Enable debugging for MPLS L2VPN connections. Output similar to the following example is generated when these tasks are performed:
- Connect three routers that are directly connected.
- Configure MPLS, MPLS LDP and Martini settings on each router, and establish a Martini connection between two routers.
- Execute the `debugging mpls l2vpn connection` command on one PE router.
- Shut down the Martini-enabled interface that is connected to the private network.

```
<PE1> debugging mpls l2vpn connection
<PE1> terminal monitor
<PE1> terminal debugging
<PE1> system-view
[PE1] interface vlan-interface 80
[PE1-Vlan-interface80] shutdown

%Jul  7 08:57:01:609 2006 PE1 IFNET/5/LINK UPDOWN:
  Vlan-interface80: link status is DOWN
*0.57830687 PE1 L2V/8/DBG:
  ! L2VPN LDP VC state DOWN:(vcid=100,vctype=5,dest=33.33.33.33)

// L2VPN LDP VC was in DOWN state.
[PE1-Vlan-interface80] display debug
MPLS L2VPN connection debugging is on
```

# Enable debugging for MPLS L2VPN errors. Output similar to the following example is generated when these tasks are performed:
- Connect three routers that are directly connected.
- Configure MPLS, MPLS LDP and CCC settings on each router, and establish a remote CCC connection between two PE routers.
- Execute the `debugging mpls l2vpn error` command on one PE router.
- Disable the MPLS function on the interface that is connected to the public network.

```
[PE1-Vlan-interface80]
[PE1-Vlan-interface80] display debugging
MPLS L2VPN error debugging is on
[PE1-Vlan-interface80] undo shutdown
```
Vlan-interface80: link status is UP

interface Vlan-interface10
ip address 100.1.1.1 255.255.255.0
mpls
mpls ldp

// Invalid remote interface was not able to get the outgoing interface index.

# Enable debugging for MPLS L2VPN events. Output similar to the following example is generated when these tasks are performed:

- Connect three routers that are directly connected.
- Configure MPLS, MPLS LDP and Martini settings on each router, and establish a Martini connection between two PE routers.
- Execute the debugging mpls l2vpn event command on one PE router.
- Disable the MPLS function on the interface that is connected to the private network.

<table>
<thead>
<tr>
<th>Transport Client</th>
<th>VC</th>
<th>Local</th>
<th>Remote</th>
<th>Tunnel</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC ID</td>
<td>Intf</td>
<td>State</td>
<td>VC Label</td>
<td>VC Label</td>
</tr>
</tbody>
</table>

PE1 display mpls l2vc
total ldp vc : 1   1 up    0 down
PE1>dis debug
MPLS L2VPN event debugging is on
[PE1-Vlan-interface80] shutdown

%Jul 10 10:01:54:640 2006 PE1 IFNET/5/LINK UPDOWN:
Vlan-interface80: link status is DOWN
*0.1196640 PE1 L2V/8/DBG:
! L2VPN - interface status change event received: interface Vlan-interface80 down

// L2VPN received interface status change event that interface VLAN-interface 80 was down.
*0.1196640 PE1 L2V/8/DBG:
! notify Final Status, IFINDEX = 120717391, ulslot = 0, ulSeqNo = -2147483613

// L2VPN notified the final status that interface index was 120717391, slot number was 0, and sequence number was -2147483613.
*0.1196703 PE1 L2V/8/DBG:
!L2VPN LDP-Downloaded FIB to MFW: Event = 2, Entry Type = 1:

// L2VPN LDP downloaded FIB to MFW. Event 2 was deleting forwarding entries, and entry type 1 indicates that the entry was received by a remote connection.
*0.1196703 PE1 L2V/8/DBG:
!L2VPN LDP-Downloaded FIB to MFW: Event = 2, Entry Type = 3:

// L2VPN LDP downloaded FIB to MFW. Event 2 was deleting forwarding entries, and entry type 3 indicates that the entry was sent from a remote connection.
*0.1196703 PE1 L2V/8/DBG:
! L2VPN - intf down: ldp intf down

// L2VPN received the interface down event indicating that the interface enabled with LDP went down.
*0.1196703 PE1 L2V/8/DBG:
! L2VPN - intf Down event receive&process

// L2VPN received and processed the interface down event.
*0.1196703 PE1 L2V/8/DBG:
! L2VPN - intf Notify Msg to IFNET process

// L2VPN notified IFNET process of the message.
*0.1196718 PE1 L2V/8/DBG:
g_ulL2vpnMsg2MFWCounter:2

// Layer 2 forwarding entry counter was 2.
*0.1196718 PE1 L2V/8/DBG:
g_ulL2vpnMsg2LDPCounter:1

// L2VPN to LDP message counter is 1.

# Enable debugging for MPLS L2VPN hot standby. Output similar to the following example is generated when these tasks are performed:

- Connect three routers that are directly connected.
- Configure MPLS, MPLS LDP and Martini settings on each router, and establish a Martini connection between two PE routers.
- Execute the `debugging mpls l2vpn event` command on one PE router.
- Shut down the interface that is connected to the private network.

```
[PE1] interface vlan-interface 80
[PE1-Vlan-interface80]
[PE1-Vlan-interface80] display this
#
interface Vlan-interface80
  shutdown
  ip address 100.5.1.2 255.255.255.0
  mpls l2vc 33.33.33.33 100
#
return
[PE1-Vlan-interface80] shutdown
  Interface Vlan-interface80 has already been shut down
[PE1-Vlan-interface80] undo shutdown
[PE1-Vlan-interface80]
%Jul 10 10:10:10:687 2006 PE1 IFNET/5/LINK UPDOWN:
  Vlan-interface80: link status is UP
*0.1686718 PE1 L2V/8/DBG:
  L2VPN CCCEncapIntf RealTime Backup Modify IfState....entered
// L2VPN interface encapsulation state change realtime backup began.
*0.1686718 PE1 L2V/8/DBG:
  L2VPN CCCEncapIntf RealTime Backup Modify IfState....leaving
// L2VPN interface encapsulation state change realtime backup finished.
*0.1686734 PE1 L2V/8/DBG:
  L2VPN LocalLdp Realtime backup Modify VCState....entered
// L2VPN local LDP information realtime backup began.
*0.1686906 PE1 L2V/8/DBG:
  L2VPN Restore Scheduler....entered
// L2VPN restore began.
*0.1686906 PE1 L2V/8/DBG:Slot=7;
  L2VPN Restore Buffer....entered
// L2VPN restore buffer began.
*0.1686906 PE1 L2V/8/DBG:Slot=7;
  L2VPN CCCEncapIntfDLL Restore....entered
// L2VPN interface encapsulation restore began.
*0.1686906 PE1 L2V/8/DBG:Slot=7;
  L2VPN CCCEncapIntfDLL Restore....leaving
```
// L2VPN interface encapsulation restore finished.
*0.1686906 PE1 L2V/8/DBG:Slot=7;
   L2VPN Parse Restore Buffer....leaving

// L2VPN parse restore buffer finished.
*0.1686922 PE1 L2V/8/DBG:Slot=7;
   L2VPN Restore Buffer....leaving

// L2VPN restore buffer finished.
*0.1686922 PE1 L2V/8/DBG:Slot=7;
   L2VPN Restore Scheduler....leaving

// L2VPN restore buffer finished.
*0.1686922 PE1 L2V/8/DBG:Slot=7;
   L2VPN Restore Scheduler....entered

// L2VPN restore began.
*0.1686922 PE1 L2V/8/DBG:Slot=7;
   L2VPN Restore Buffer....entered

// L2VPN restore buffer began.
*0.1686937 PE1 L2V/8/DBG:Slot=7;
   L2VPN Parse Restore Buffer....entered

// L2VPN parse restore buffer began.
*0.1686937 PE1 L2V/8/DBG:Slot=7;
   L2VPN LocalLdp Restore....entered

// L2VPN local LDP information restore began.
*0.1686937 PE1 L2V/8/DBG:Slot=7;
   L2VPN LocalLdp Restore....leaving

// L2VPN local LDP information restore finished.
*0.1686937 PE1 L2V/8/DBG:Slot=7;
   L2VPN Parse Restore Buffer....leaving

// L2VPN parse restore buffer finished.
*0.1686937 PE1 L2V/8/DBG:Slot=7;
   L2VPN Restore Buffer....leaving

// L2VPN restore buffer finished.
*0.1686937 PE1 L2V/8/DBG:Slot=7;
   L2VPN Restore Scheduler....leaving

// L2VPN restore finished.
MPLS L3VPN debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

depending l3vpn

Use `debugging l3vpn` to enable debugging for L3VPN.

Use `undo debugging l3vpn` to disable debugging for L3VPN.

Syntax

```
debugging l3vpn { all | hsb | cfgmsg }
undo debugging l3vpn { all | hsb | cfgmsg }
```

Default

The debugging for L3VPN is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `all`: Specifies all types of debugging for L3VPN.
- `hsb`: Specifies debugging for L3VPN hot standby.
- `cfgmsg`: Specifies debugging for L3VPN configuration message processing.

Usage guidelines

Table 1 describes the output fields and messages for the `debugging l3vpn` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3VPN VRFMGM VrfCreate....entered</td>
<td>L3VPN began to create VPN instances.</td>
</tr>
<tr>
<td>L3VPN VRFMGM VrfCreate....leaving</td>
<td>L3VPN finished creating VPN instances.</td>
</tr>
<tr>
<td>L3VPN VRFMGM RdAssign....entered</td>
<td>L3VPN began to assign a route distinguisher.</td>
</tr>
<tr>
<td>L3VPN VRFMGM RdAssign....leaving</td>
<td>L3VPN finished assigning a route distinguisher.</td>
</tr>
<tr>
<td>L3VPN VRFMGM RtList Assign....entered</td>
<td>L3VPN began to assign VPN targets.</td>
</tr>
<tr>
<td>L3VPN VRFMGM RtList Assign....leaving</td>
<td>L3VPN finished assigning VPN targets.</td>
</tr>
<tr>
<td>L3VPN VRFMGM CreateDesc....entered</td>
<td>L3VPN began to create a description for the current VPN instance.</td>
</tr>
<tr>
<td>L3VPN VRFMGM CreateDesc....leaving</td>
<td>L3VPN created a description for the current VPN instance.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>L3VPN VRFMGM DeleteDesc....entered</td>
<td>L3VPN began to delete the description of the current VPN instance.</td>
</tr>
<tr>
<td>L3VPN VRFMGM DeleteDesc....leaving</td>
<td>L3VPN deleted the description of the current VPN instance.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Import Route Policy Assign....entered</td>
<td>L3VPN began to assign an import route policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Import Route Policy Assign....leaving</td>
<td>L3VPN finished assigning an import route policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM MaxRoute....entered</td>
<td>L3VPN started the limit on the maximum number of VPN routes.</td>
</tr>
<tr>
<td>L3VPN VRFMGM MaxRoute....leaving</td>
<td>L3VPN ended the limit on the maximum number of VPN routes.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Tunnel Policy Assign....entered</td>
<td>L3VPN began to assign a tunnel policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Tunnel Policy Assign....leaving</td>
<td>L3VPN finished assigning a tunnel policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Export Route Policy Assign....entered</td>
<td>L3VPN began to assign an export route policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Export Route Policy Assign....leaving</td>
<td>L3VPN finished assigning an export route policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Import Route Policy Delete....entered</td>
<td>L3VPN began to delete an import route policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Import Route Policy Delete....leaving</td>
<td>L3VPN finished deleting an import route policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Export Route Policy Delete....entered</td>
<td>L3VPN began to delete an export route policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Export Route Policy Delete....leaving</td>
<td>L3VPN finished deleting an export route policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Tunnel Policy Delete....entered</td>
<td>L3VPN began to delete a tunnel policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM Tunnel Policy Delete....leaving</td>
<td>L3VPN finished deleting a tunnel policy.</td>
</tr>
<tr>
<td>L3VPN VRFMGM RtDelete.....entered</td>
<td>L3VPN began to delete a VPN-Target.</td>
</tr>
<tr>
<td>L3VPN VRFMGM RtDelete.....leaving</td>
<td>L3VPN finished deleting a VPN-Target.</td>
</tr>
<tr>
<td>L3VPN VRFMGM RtDelete All.....entered</td>
<td>L3VPN began to delete a route distinguisher.</td>
</tr>
<tr>
<td>L3VPN VRFMGM RtDelete All.....leaving</td>
<td>L3VPN finished deleting a route distinguisher.</td>
</tr>
<tr>
<td>L3VPN VRFMGM IfAssociate.....entered</td>
<td>L3VPN began to associate a VPN with an interface.</td>
</tr>
<tr>
<td>L3VPN VRFMGM IfAssociate.....leaving</td>
<td>L3VPN finished associating a VPN with an interface.</td>
</tr>
<tr>
<td>L3VPN VRFMGM IfDisassociate.....entered</td>
<td>L3VPN began to disassociate a VPN from an interface.</td>
</tr>
<tr>
<td>L3VPN VRFMGM IfDisassociate.....leaving</td>
<td>L3VPN finished disassociating a VPN from an interface.</td>
</tr>
<tr>
<td>L3VPN VRFMGM VrfDelete.....entered</td>
<td>L3VPN began to delete a VPN.</td>
</tr>
<tr>
<td>L3VPN VRFMGM VrfDelete.....leaving</td>
<td>L3VPN finished deleting a VPN.</td>
</tr>
<tr>
<td>IntfDelete Notify.....entered</td>
<td>L3VPN began to delete an interface (logical).</td>
</tr>
<tr>
<td>IntfDelete Notify.....leaving</td>
<td>L3VPN finished deleting an interface (logical).</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>L3VPN CORE Get Batch BackUp Data....entered</td>
<td>L3VPN began to get batch backup data.</td>
</tr>
<tr>
<td>L3VPN CORE Get Batch BackUp Data....leaving</td>
<td>L3VPN finished getting batch backup data.</td>
</tr>
<tr>
<td>PackOneBatchVrf....entered</td>
<td>L3VPN began to get a batch of VRFs.</td>
</tr>
<tr>
<td>PackOneBatchVrf....leaving</td>
<td>L3VPN finished getting a batch of VRFs.</td>
</tr>
<tr>
<td>PackOneVrf....entered</td>
<td>L3VPN began to get one VRF.</td>
</tr>
<tr>
<td>PackOneVrf....leaving</td>
<td>L3VPN finished getting one VRF.</td>
</tr>
<tr>
<td>Vrf: VrfId bind Interface: IfIndex success</td>
<td>L3VPN bound a VRF with an interface successfully.</td>
</tr>
<tr>
<td>Vrf: VrfId unbind Interface: IfIndex success</td>
<td>L3VPN unbound a VRF from an interface successfully.</td>
</tr>
<tr>
<td>Vrf: VrfId bind Interface: IfIndex failure</td>
<td>L3VPN failed to bind a VRF with an interface.</td>
</tr>
<tr>
<td>Vrf: VrfId unbind Interface: IfIndex failure</td>
<td>L3VPN failed to unbind a VRF from an interface.</td>
</tr>
<tr>
<td>Receive (chassis chassisID, slot slotID) Plug In Notify</td>
<td>L3VPN received the card (in chassis chassisID, slot slotID) plug-in notify message.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable all types of debugging for L3VPN. Output similar to the following example is generated when these tasks are performed in order:

- Create a VPN instance.
- Configure an RD and VPN target.
- Configure a description for the VPN instance and delete it.
- Configure an import route policy.
- Configure a VPN route number limit.
- Bind the VPN instance to an interface.

```
<Sysname> debugging l3vpn all
[Sysname] ip vpn-instance vpna
*0.1362469 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM VrfCreate....entered
  // L3VPN began to create a VPN instance.
*0.1362469 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM VrfCreate....leaving
  // L3VPN finished creating the VPN instance.
[Sysname-vpn-instance-vpna] route-distinguisher 100:1
*0.1538078 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM RdAssign....entered
  // L3VPN began to assign an RD.
*0.1538078 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM RdAssign....leaving
  // L3VPN finished assigning the RD.
[Sysname-vpn-instance-vpna] vpn-target 111:1
IVT Assignment result:
VPN-Target assignment is successful
EVT Assignment result:
```
VPN-Target assignment is successful
*0.1669656 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM RtList Assign....entered
// L3VPN began to assign VPN targets.
*0.1669656 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM RtList Assign....leaving
// L3VPN finished assigning VPN targets.
*0.1669656 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM RtList Assign....entered
// L3VPN began to assign VPN target.
*0.1669656 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM RtList Assign....leaving
// L3VPN finished assigning VPN target.
[Sysname-vpn-instance-vpna]description this is orange's vpn
*0.1912719 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM CreateDesc....entered
// L3VPN began to create a description for the current VPN instance.
*0.1912719 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM CreateDesc....leaving
// L3VPN created a description for the current VPN instance.
[Sysname-vpn-instance-vpna]undo description
*0.3361156 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM DeleteDesc....entered
// L3VPN began to delete the description of the current VPN instance.
*0.3361156 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM DeleteDesc....leaving
// L3VPN deleted the description of the current VPN instance.
[Sysname-vpn-instance-vpna]import route-policy irp1
Policy configuration is successful, but needs to be created
*0.2217297 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM Import Route Policy Assign....entered
// L3VPN began to assign import route policy.
*0.2217297 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM Import Route Policy Assign....leaving
// L3VPN finished assigning import route policy.
[Sysname-vpn-instance-vpna]routing-table limit 1000 100
*0.2413234 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM MaxRoute....entered
// L3VPN started the limit on the maximum number of VPN routes.
*0.2413234 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM MaxRoute....leaving
// L3VPN ended the limit on the maximum number of VPN routes.
[Sysname-vpn-instance-vpna]tnl-policy tp1
Policy configuration is successful, but needs to be created
*0.2806641 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM Tunnel Policy Assign....entered

// L3VPN began to assign tunnel policy.
*0.2806641 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM Tunnel Policy Assign....leaving

// L3VPN finished assigning tunnel policy.
[Sysname-vpn-instance-vpna]export route-policy erp1
Policy configuration is successful, but needs to be created
*0.2118578 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM Export Route Policy Assign....entered

// L3VPN began to assign export route policy.
*0.2118578 Sysname L3VPN/8/L3VPNDBG:
L3VPN VRFMGM Export Route Policy Assign....leaving

// L3VPN finished assigning export route policy.
[Sysname-Vlan-interface1]ip binding vpn-instance vpna
Vrf: 1 bind Interface: 120717312 success

// L3VPN bound the VRF to the interface.
[Sysname-Vlan-interface1]undo ip binding vpn-instance vpna
Vrf: 1 unbind Interface: 120717312 success

// L3VPN unbound the VRF from the interface.
MPLS OAM debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging mpls oam

⚠️ CAUTION:
Executing this debugging command might affect sending and receiving of MPLS OAM protocol packets. Use this command only when necessary.

Use `debugging mpls oam` to enable one or all types of debugging for MPLS OAM. Use `undo debugging mpls oam` to disable MPLS OAM debugging.

Syntax

```
debugging mpls oam { all | bdi | cv | decode | defect-detect | error | fdi | ffd |
fsm | hsb | main | packet | process | timer }
undo debugging mpls oam { all | bdi | cv | decode | defect-detect | error | fdi |
ffd | fsm | hsb | main | packet | process | timer }
```

Default

No debugging is enabled for MPLS OAM.

Views

User view

Default command level

1: Monitor level

Parameters

- `all`: Specifies all types of debugging.
- `bdi`: Specifies debugging for backward defect indication (BDI) messages.
- `cv`: Specifies debugging for connectivity verification (CV) messages.
- `decode`: Specifies debugging for encoding and decoding.
- `defect-detect`: Specifies debugging for defect detection.
- `error`: Specifies debugging for errors.
- `fdi`: Specifies debugging for forward defect indication (FDI) messages.
- `ffd`: Specifies debugging for fast fault detection.
- `fsm`: Specifies debugging for finite state machine (FSM).
- `hsb`: Specifies debugging for hot standby (HSB).
- `main`: Specifies the main debugging.
**packet**: Specifies debugging for packets.

**process**: Specifies debugging for processing.

**timer**: Specifies debugging for timers.

**Usage guidelines**

Table 1 describes the output fields and messages for the `debugging mpls oam bdi` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>OAM DD</td>
<td>OAM defect detection.</td>
</tr>
</tbody>
</table>

Table 2 describes the output fields and messages for the `debugging mpls oam cv` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>OAM DD</td>
<td>OAM defect detection.</td>
</tr>
</tbody>
</table>

Table 3 describes the output fields and messages for the `debugging mpls oam decode` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>BIP16</td>
<td>BIP16 check.</td>
</tr>
<tr>
<td>Frequency</td>
<td>Frequency of OAM packets.</td>
</tr>
<tr>
<td>padding</td>
<td>Packet payload.</td>
</tr>
<tr>
<td>defect location</td>
<td>Number of the Autonomous System (AS) where the defect is located.</td>
</tr>
<tr>
<td>Lsr id</td>
<td>LSR ID in the TTSI.</td>
</tr>
<tr>
<td>tunnel id</td>
<td>Tunnel ID in the TTSI.</td>
</tr>
<tr>
<td>Out Label</td>
<td>Outgoing label.</td>
</tr>
<tr>
<td>processing board</td>
<td>Main processing board.</td>
</tr>
<tr>
<td>InLabel</td>
<td>Incoming label.</td>
</tr>
<tr>
<td>BoardId</td>
<td>Number of the main processing board.</td>
</tr>
</tbody>
</table>

Table 4 describes the output fields and messages for the `debugging mpls oam defect-detect` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>Lsr id</td>
<td>LSR ID in the TTSI.</td>
</tr>
</tbody>
</table>
### Table 5
Table 5 describes the output fields and messages for the `debugging mpls oam error` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel id</td>
<td>Tunnel ID in the TTSI.</td>
</tr>
<tr>
<td>ASM</td>
<td>OAM state machine.</td>
</tr>
<tr>
<td>ext-bdi</td>
<td>Extended BDI, used for RLSN.</td>
</tr>
<tr>
<td>available</td>
<td>available state of the state machine.</td>
</tr>
<tr>
<td>unavailable</td>
<td>unavailable state of the state machine.</td>
</tr>
<tr>
<td>RLSN</td>
<td>Remote link status notification.</td>
</tr>
</tbody>
</table>

### Table 38
Table 38 Output from the `debugging mpls oam error` command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>ava2unava</td>
<td>OAM state machine changes from available to unavailable.</td>
</tr>
<tr>
<td>OAM DD</td>
<td>OAM defect detection.</td>
</tr>
<tr>
<td>PS</td>
<td>Protection switching.</td>
</tr>
<tr>
<td>HA</td>
<td>High availability.</td>
</tr>
<tr>
<td>BoardId</td>
<td>Number of the main processing board.</td>
</tr>
</tbody>
</table>

### Table 6
Table 6 describes the output fields and messages for the `debugging mpls oam fdi` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
</tbody>
</table>

### Table 39
Table 39 Output from the `debugging mpls oam fdi` command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
</tbody>
</table>

### Table 7
Table 7 describes the output fields and messages for the `debugging mpls oam ffd` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>OAM DD</td>
<td>OAM defect detection.</td>
</tr>
</tbody>
</table>

### Table 40
Table 40 Output from the `debugging mpls oam ffd` command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
</tbody>
</table>

### Table 8
Table 8 describes the output fields and messages for the `debugging mpls oam fsm` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>ASM</td>
<td>OAM defect detect.</td>
</tr>
<tr>
<td>ava2unava</td>
<td>OAM state machine changes from available to unavailable.</td>
</tr>
<tr>
<td>Unava</td>
<td>unavailable state of the OAM state machine.</td>
</tr>
</tbody>
</table>
Table 9 describes the output fields and messages for the `debugging mpls oam hsb` command.

**Table 42 Output from the debugging mpls oam hsb command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>status</td>
<td>HA state.</td>
</tr>
<tr>
<td>ha type</td>
<td>Type of the HA event.</td>
</tr>
</tbody>
</table>

Table 10 describes the output fields and messages for the `debugging mpls oam main` command.

**Table 43 Output from the debugging mpls oam main command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>MPLS OAM MN</td>
<td>Main debugging of MPLS OAM.</td>
</tr>
<tr>
<td>ha type</td>
<td>Type of the HA event.</td>
</tr>
<tr>
<td>InLabel</td>
<td>Incoming label.</td>
</tr>
<tr>
<td>BoardId</td>
<td>Number of the main processing board.</td>
</tr>
<tr>
<td>tunnel id</td>
<td>Tunnel ID of the ingress.</td>
</tr>
<tr>
<td>Lsp Id</td>
<td>LSP ID of the CRLSP.</td>
</tr>
<tr>
<td>InLabel</td>
<td>Incoming label of the static LSP.</td>
</tr>
<tr>
<td>TnlIfindex</td>
<td>Tunnel interface index.</td>
</tr>
<tr>
<td>Lsp Status</td>
<td>Status of the CRLSP.</td>
</tr>
<tr>
<td>IfIndex</td>
<td>Interface index.</td>
</tr>
<tr>
<td>MsgType</td>
<td>Interface message.</td>
</tr>
<tr>
<td>rev lsp</td>
<td>Backward LSP.</td>
</tr>
<tr>
<td>pdt</td>
<td>Product.</td>
</tr>
</tbody>
</table>

Table 11 describes the output fields and messages for the `debugging mpls oam packet` command.

**Table 44 Output from the debugging mpls oam packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>Processing Board</td>
<td>Number of the main processing board.</td>
</tr>
<tr>
<td>TTSI-Lsrid</td>
<td>LSR ID in the TTSI.</td>
</tr>
<tr>
<td>TTSI-Tunnelid</td>
<td>Tunnel ID in the TTSI.</td>
</tr>
</tbody>
</table>

Table 12 describes the output fields and messages for the `debugging mpls oam process` command.
Table 45 Output from the debugging mpls oam process command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM Index</td>
<td>OAM instance index.</td>
</tr>
<tr>
<td>release oam</td>
<td>OAM sent the OAM instance to the main processing board.</td>
</tr>
<tr>
<td>update entry config</td>
<td>OAM updated the OAM instance configuration.</td>
</tr>
</tbody>
</table>

Table 13 describes the output fields and messages for the `debugging mpls oam timer` command.

Table 46 Output from the debugging mpls oam timer command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM ava2unava timer</td>
<td>available state timer of the OAM state machine.</td>
</tr>
<tr>
<td>ASM unava2ava timer</td>
<td>unavailable state timer of the OAM state machine.</td>
</tr>
<tr>
<td>Rlsn timer</td>
<td>Timer for sending extended BDI messages.</td>
</tr>
<tr>
<td>release timer</td>
<td>OAM released the OAM instance timer.</td>
</tr>
</tbody>
</table>

Examples

# Enable MPLS OAM processing debugging. Output similar to the following example is generated when these tasks are performed:

- Configure a TE tunnel between two routers.
- Execute the `debugging mpls oam process` command on the tunnel ingress router.
- Configure an OAM instance on the router.

```
<PE1> debugging mpls oam process
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:  
17:6571: begin to find oam by detect tunnel
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:  
17:6581: find oam by tunnel ifindex at ingress
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:  
17:6618: find oam from detect tunnel hash table
    // OAM was finding the OAM instance by the detected tunnel.
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:  
17:6632: have not find oam from detect tunnel hash table
    // OAM instance was not found.
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:  
17:4473: begin to add oam entry.
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:  
17:4498: Success to alloc oam entry from gads.
    // OAM started to add the OAM instance.
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:  
17:4544: begin to fill oam entry.
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:  
17:5757: begin to add detect tunnel hash node
    // OAM filled information into the OAM instance.
```
// OAM started to send the OAM instance to the main processing board.
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:
17:4881: Begin to release oam!

// OAM started to select the main processing board.
*Nov 29 16:01:18:125 2006 PE1 OAM/7/PROCESS:
17:9258: GetProcBoard:begin to select process board
*Nov 29 16:01:18:146 2006 PE1 OAM/7/PROCESS:
17:9474: SelAgtBoard process start

// OAM was obtaining the board where the outgoing interface resides according to the ARP information.
*Nov 29 16:01:18:162 2006 PE1 OAM/7/PROCESS:
17:13291: GetBoardByArp: ulVrfIndex = 0, ulNextHop = c010102, ulIfIndex = 120717313

// Board 1 was obtained.
*Nov 29 16:01:18:365 2006 PE1 OAM/7/PROCESS:
17:2649: begin to add search node
*Nov 29 16:01:18:365 2006 PE1 OAM/7/PROCESS:
17:2671: begin to select which search node to add
*Nov 29 16:01:18:365 2006 PE1 OAM/7/PROCESS:
17:2694: oam proc:Add ttsi node at ingressmpl
*Nov 29 16:01:18:831 2006 PE1 OAM/7/PROCESS:
17:2897: begin to add ttsi node
*Nov 29 16:01:18:831 2006 PE1 OAM/7/PROCESS:
17:2916: sucess to memory alloc for ttsi node
*Nov 29 16:01:18:831 2006 PE1 OAM/7/PROCESS:
17:2947: add ttsi node at ingress table
*Nov 29 16:01:18:831 2006 PE1 OAM/7/PROCESS:
17:3046: finish to add ttsi node
*Nov 29 16:01:18:831 2006 PE1 OAM/7/PROCESS:
17:2785: finish to add search node
*Nov 29 16:01:18:831 2006 PE1 OAM/7/PROCESS:
17:4717: OAM_SetEntryStatRecord: ulIgrReleaseEntryNumLowFreq add 1

// OAM added the search table.
*Nov 29 16:01:18:847 2006 PE1 OAM/7/PROCESS:Slot=1;
17:1580:
IPC Msg Type: 4

// The interface board received a message of type 4 (search table).
Nov 29 16:01:18:847 2006 PE1 OAM/7/PROCESS:Slot=1; 17:2649: begin to add search node
Nov 29 16:01:18:847 2006 PE1 OAM/7/PROCESS:Slot=1; 17:2671: begin to select which search node to add
Nov 29 16:01:18:847 2006 PE1 OAM/7/PROCESS:Slot=1; 17:2694: oam proc:Add ttsi node at ingress
Nov 29 16:01:18:864 2006 PE1 OAM/7/PROCESS:Slot=1; 17:2897: begin to add ttsi node
Nov 29 16:01:18:879 2006 PE1 OAM/7/PROCESS:Slot=1; 17:2916: sucess to memory alloc for ttsi node
Nov 29 16:01:18:879 2006 PE1 OAM/7/PROCESS:Slot=1; 17:2947: add ttsi node at ingress table
Nov 29 16:01:18:879 2006 PE1 OAM/7/PROCESS:Slot=1; 17:3046: finish to add ttsi node
Nov 29 16:01:18:879 2006 PE1 OAM/7/PROCESS:Slot=1; 17:2785: finish to add search node

// OAM added a search node.
Nov 29 16:01:18:879 2006 PE1 OAM/7/PROCESS:Slot=1; 17:1580:
IPC Msg Type: 1
// The interface board received a message of type 1 (add instance).
Nov 29 16:01:18:879 2006 PE1 OAM/7/PROCESS:Slot=1; 17:0060: begin to set entry
Nov 29 16:01:18:879 2006 PE1 OAM/7/PROCESS:Slot=1; 17:0072: seccess to alloc gads node
Nov 29 16:01:18:895 2006 PE1 OAM/7/PROCESS:Slot=1; 17:0121: MPLS OAM SetEntryIO: start to mallo the space for oam packet
Nov 29 16:01:18:895 2006 PE1 OAM/7/PROCESS:Slot=1; 17:0140: MPLS OAM SetEntryIO: Start to fill the CV FFD Packet s
Nov 29 16:01:18:908 2006 PE1 OAM/7/PROCESS:Slot=1; 17:0153: MPLS OAM SetEntryIO: End to fill the CV FFD Packet

// The interface board started to add an OAM instance.
Nov 29 16:01:18:908 2006 PE1 OAM/7/PROCESS:Slot=1; 17:0188: MPLS OAM SetEntryIO: sucess to malloc the space for oam packet and fill the packet

// The OAM instance was added successfully.
Nov 29 16:01:18:908 2006 PE1 OAM/7/PROCESS:Slot=1; 17:0221: finish to add entry on io board oam ingress enable Tunnel1

// OAM finished adding the OAM instance.
MPLS TE debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging isis traffic-eng

Use debugging isis traffic-eng to enable debugging for IS-IS traffic engineering.

Use undo debugging isis traffic-eng to disable debugging for IS-IS traffic engineering.

Syntax

debugging isis traffic-eng { advertisement | event } [ process-id | vpn-instance vpn-instance-name ]
undo debugging isis traffic-eng { advertisement | event } [ process-id | vpn-instance vpn-instance-name ]

Default

Debugging for IS-IS traffic engineering is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

advertisement: Specifies debugging for IS-IS advertisements.

event: Specifies debugging for IS-IS traffic engineering events.

process-id: Specifies debugging for an IS-IS process. The process-id argument represents the IS-IS process ID in the range of 1 to 65535.

vpn-instance vpn-instance-name: Specifies debugging for an MPLS L3VPN instance. The vpn-instance-name argument represents the VPN instance name, a case-sensitive string of 1 to 31 characters.

Examples

# Enable debugging for IS-IS traffic engineering events.
<Sysname> debugging isis traffic-eng event

debugging mpls moam packet

Use debugging mpls moam packet to enable debugging for MPLS TP OAM (MOAM).

Use undo debugging mpls moam packet to disable debugging for MOAM.

Syntax

debugging mpls moam packet
undo debugging mpls moam packet
Default

Debugging for MOAM is disabled.

Views

User view

Default command level

1: Monitor level

Examples

# Enable debugging for MOAM packets. Use the moam lm command to detect the packet loss rates for a bidirectional MPLS TE tunnel on the device.
<Sysname> debugging mpls moam packet
<Sysname> moam lm -c 1 te tunnel 2
Loss measurement for Tunnel 2, press CTRL_C to break
*Jan 30 16:46:36:781 2011 Sysname MOAM/7/Packet:
>>>Send LM query msg.
Sequence number: 0

// MOAM sent an LM request message. The sequence number of the message is 0.
*Jan 30 16:46:37:062 2011 Sysname MOAM/7/Packet:
>>>Send LM query msg.
Sequence number: 1

// MOAM sent an LM request message. The sequence number of the message is 1.
  1. TxLoss = 1 (100.00% loss), Rxloss = 1 (100.00% loss)
*Jan 30 16:46:37:094 2011 Sysname MOAM/7/Packet:
Received LM packet: C1: 0, C2: 0, C3: 0, C4: 0
*Jan 30 16:46:37:125 2011 Sysname MOAM/7/Packet:
Calculate send packet loss
Ingress send   : 1
Egress  receive: 0
Txloss         : 1

// One sent packet was lost.
*Jan 30 16:46:37:125 2011 Sysname MOAM/7/Packet:
Calculate receive packet loss
Egress  send   : 1
Ingress receive: 0
Rxloss         : 1

// One packet failed to be received.
--- Loss measurement statistics for Tunnel 2 ---
   TxLoss/TxTotal = 1/1 (100.00% loss)
   RxLoss/RxTotal = 1/1 (100.00% loss)

debugging mpls rsvp-te

Use debugging mpls rsvp-te to enable debugging for MPLS RSVP-TE.
Use undo debugging mpls rsvp-te to disable debugging for MPLS RSVP-TE.
Syntax

d debugging mpls rsvp-te { all | authentication | bundle | encdec | error | hello | hsb | main | msg-hex | path | perr | ptear | rconf | rerr | resv | rtear | socket | srefresh | timer | tool | traffic-control | tunnel-id { tunnel-id | all } }

undo debugging mpls rsvp-te { all | authentication | bundle | encdec | error | hello | hsb | main | msg-hex | path | perr | ptear | rconf | rerr | resv | rtear | socket | srefresh | timer | tool | traffic-control | tunnel-id { tunnel-id | all } }

Default

Debugging for MPLS RSVP-TE is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all: Specifies all types of debugging.
authentication: Specifies debugging for authentication.
bundle: Specifies debugging for bundle.
encdec: Specifies debugging for encoding and decoding.
error: Specifies debugging for errors.
hello: Specifies debugging for hello.
hsb: Specifies debugging for hot standby backup.
main: Specifies debugging for main.
msg-hex: Specifies debugging for message hex format.
path: Specifies debugging for paths.
perr: Specifies debugging for path errors.
ptear: Specifies debugging for path tearing down.
rconf: Specifies debugging for reservation confirmation.
rerr: Specifies debugging for reservation errors.
resv: Specifies debugging for reservation message.
rtear: Specifies debugging for reservation tearing down.
socket: Specifies debugging for socket.
srefresh: Specifies debugging for Srefresh.
timer: Specifies debugging for timer.
tool: Specifies debugging for tools.
traffic-control: Specifies debugging for traffic control.
tunnel-id: Specifies tunnel debugging. The tunnel-id argument specifies a single tunnel by ID, which is in the range of 0 to 2048. The all keyword specifies all tunnels.
### Usage guidelines

For the completeness of debugging information, H3C recommends that you enable `debugging mpls rsvp-te tunnel-id` for each type of RSVP-TE debugging except for `debugging mpls rsvp-te all`.

Table 1 describes output fields and messages for the `debugging mpls rsvp authentication` command.

#### Table 47: Output from the debugging mpls rsvp authentication command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local IfEntry number is x.</td>
<td>The local RSVP-TE interface sequence number is X.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the `debugging mpls rsvp error` command.

#### Table 48: Output from the debugging mpls rsvp error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid IfIndex</td>
<td>Invalid interface index.</td>
</tr>
<tr>
<td>IF Not Enbale Authenticate</td>
<td>Authentication is not enabled on the interface.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the `debugging mpls rsvp hello` command.

#### Table 49: Output from the debugging mpls rsvp hello command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nbr is lost, process Frr Nbr lost</td>
<td>Neighbor was lost. Frr processed neighbor loss.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the `debugging mpls rsvp hsb` command.

#### Table 50: Output from the debugging mpls rsvp hsb command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMB: Backup Global Data and send to HAC</td>
<td>Active MPU: Backed up the global data and sent it to HAC.</td>
</tr>
<tr>
<td>SMB: Restore Global Data</td>
<td>Standby MPU: Restored the global data.</td>
</tr>
<tr>
<td>SMB: RealTime Restore TCSB Data</td>
<td>Standby MPU: Restored the traffic control state block (TCSB) data.</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the `debugging mpls rsvp main` command.

#### Table 51: Output from the debugging mpls rsvp main command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSVP GR process: RSVP NBR x.x.x.x restarting.</td>
<td>RSVP GR process: RSVP neighbor xxx is restarting.</td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the `debugging mpls rsvp path` command.

#### Table 52: Output from the debugging mpls rsvp path command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path mesg Received with Lih xxx for IfAddr xxx</td>
<td>MPLS TE received a path message with the logical interface handle (Lih) of xxx from interface xxx.</td>
</tr>
</tbody>
</table>
Unable to add PSB in IfEntry link  
MPLS TE was unable to add PSB to the RSVP-TE interface information.

Table 7 describes output fields and messages for the **debugging mpls rsvp ptear** command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TearRSB by OutIf filter is Null, Send PathTear Msg</td>
<td>When processing TearRSB by outbound interface, MPLS TE found the filter is null and sent a PathTear message.</td>
</tr>
</tbody>
</table>

Table 8 describes output fields and messages for the **debugging mpls rsvp resv** command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can't get InClF</td>
<td>MPLS TE failed to get the inbound interface.</td>
</tr>
<tr>
<td>Resv mesg Received with Lih xxx for IfAdd xxx</td>
<td>Resv message received with logical interface handle xxx for interface address xxx.</td>
</tr>
</tbody>
</table>

Table 9 describes output fields and messages for the **debugging mpls rsvp traffic-control** command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start update Tc process</td>
<td>MPLS TE started update traffic processing.</td>
</tr>
</tbody>
</table>

Examples

The following examples are based on the scenario in which two directly-connected devices have been enabled with MPLS basic capability, MPLS TE basic capability, and MPLS RESV-TE capability.

# Enable RSVP-TE authentication on the interfaces between the two devices. Execute the **debugging mpls rsvp-te authentication** and **debugging mpls rsvp-te tunnel-id all** commands. Then the two devices establish an MPLS TE tunnel.

```
<Sysname> debugging mpls rsvp-te authentication
<Sysname> debugging mpls rsvp-te tunnel-id all
  // RSVP sent a sequence number 1163413641 2.
*Nov 13 10:29:41:444 2006 Sysname RSVP/7/AUTH: 04:0278: Local IfEntry number is 1163413640 2.
  // The sequence number of the local RSVP-TE interface is 1163413640 2.
  // RSVP received a sequence number 1163413640 3.
```
Enable the RSVP-TE Hello function in MPLS view and interface view. Execute both the `debugging mpls rsvp-te hello` and `debugging mpls rsvp-te tunnel-id all` commands. Then the two devices establish an MPLS TE tunnel.

```
<Sysname> debugging mpls rsvp-te hello
<Sysname> debugging mpls rsvp-te tunnel-id all
```

```
0:0353: Receiving Hello REQ Message form 80.1.1.2:
// RSVP received a Hello REQ message from 80.1.1.2.
```

```
*Nov 13 10:39:00:618 2006 Sysname RSVP/7/HELLO:
0:0359: Src_Instance : 0x1d1  Dst_Instance : 0x1ce
// The source instance is 0x1d1, and the destination instance is 0x1ce.
```

```
0:0506: Sending Hello ACK Message:
```

```
*Nov 13 10:39:00:634 2006 Sysname RSVP/7/HELLO:
0:0511: Src_Instance : 0x1ce  Dst_Instance : 0x1d1
// The source instance is 0x1ce, and the destination instance is 0x1d1.
```

Execute both the `debugging mpls rsvp-te path` and `debugging mpls rsvp-te tunnel-id all` commands. Then the two devices establish an MPLS TE tunnel.

```
<Sysname> debugging mpls rsvp-te path
<Sysname> debugging mpls rsvp-te tunnel-id all
```

```
*Nov 13 10:46:19:130 2006 Sysname RSVP/7/PATH:
13:2566: Form and Send Path Msg!
// RSVP formed and sent a PATH message.
```

```
*Nov 13 10:46:19:145 2006 Sysname RSVP/7/PATH:
13:2808: Path msg sent with Lih 2818064 for IfAdd 50010101
// RSVP sent a PATH message from interface 50010101, with the LIH of 2818064.
```

```
*Nov 13 10:46:19:145 2006 Sysname RSVP/7/PATH:
13:5017: Successfully send path refresh message...
// RSVP sent a PATH refresh message.
```

```
*Nov 13 10:46:19:145 2006 Sysname RSVP/7/PATH:
13:5023: Tunnel address = 88.2.2.2, Tunnel ID =1, next hop = 80.1.1.2
// The tunnel’s destination address is 88.2.2.2, the tunnel ID is 1, and the next hop is 80.1.1.2.
```

```
*Nov 13 10:46:19:145 2006 Sysname RSVP/7/PATH:
13:5028: Timer interval configured=30000 : Timer interval used =29100
// The refresh timer setting is 30000 milliseconds, and the actual timer value is 29100 milliseconds.
```

```
*Nov 13 10:46:19:145 2006 Sysname RSVP/7/PATH:
13:5031: PATH msg MsgId = 0x2
// The MsgId of the PATH message is 0x2.
```

Configure FRR so that when the PLR node detects a need for FRR switchover, it will send a path error message to notify the upstream node of the FRR switchover. Execute both the `debugging mpls rsvp-te perr` and `debugging mpls rsvp-te tunnel-id all` commands.

```
<Sysname> debugging mpls rsvp-te perr
<Sysname> debugging mpls rsvp-te tunnel-id all
```
RSVP started to send a PathErr message.
RSVP finished sending the PathErr message.

Execute both the `debugging mpls rsvp-te ptear` and `debugging mpls rsvp-te tunnel-id all` commands, and then delete the tunnel.

LSPM deleted the LSP and is preparing to send a PathTear message.
RSVP constructed and sent the PathTear message.
RSVP deleted the FiltSpec in RSB.
RSVP deleted PSB.

Execute both the `debugging mpls rsvp-te resv` and `debugging mpls rsvp-te tunnel-id all` commands. Then the two devices establish an MPLS TE tunnel.

RSVP started RESV message processing.
RSVP received from address 50030101 a RESV message with Lih 2818064.
The session is 88.1.1.1:6.
The reservation style is SE.
RSVP started to process the current flow descriptor.
RSVP found a matching PSB.
// RSVP updated the old RSB.

// The bandwidth request carried by the message is 0.

// The LSP ID is 6.

// RSVP finished displaying LSP ID.

// RSVP got the PSB for FilterSpec.

RRO in RSB: 80.3.1.1  
Label in RSB: 1037  
RRO in RSB: 88.3.3.3  
RRO in RSB: 80.2.1.2  
RRO in RSB: 80.2.1.1  
Label in RSB: 1033  
RRO in RSB: 88.2.2.2  
RRO in RSB: 80.1.1.2  
RRO in RSB: 80.1.1.1  
Label in RSB: 3  
RRO in RSB: 88.1.1.1  
*Nov 13 11:55:30:00 2006 Sysname RSVP/7/RESV: 19:0863: RESV msg MsgID = 0x0  
// The Msg ID of the RESV message is 0x0.

*Nov 13 11:55:30:00 2006 Sysname RSVP/7/RESV: 19:0984: Delete Message ID  
// RSVP deleted the message ID.

// RSVP finished processing the current flow descriptor.

// RSVP finished processing the RESV message.

debugging mpls te cspf

Use `debugging mpls te cspf` to enable debugging for CSPF.  
Use `undo debugging mpls te cspf` to disable debugging for CSPF.
Syntax

depth mpls te cspf { all | computation | errors | events | tedb }
undo debugging mpls te cspf { all | computation | errors | events | tedb }

Default

Debugging for CSPF is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all: Specifies all types of debugging for MPLS TE CSPF.
computation: Specifies debugging for CSPF calculation.
error: Specifies debugging for errors.
event: Specifies debugging for events.
tedb: Specifies debugging for traffic engineering database (TEDB).

Usage guidelines

Table 10 describes output fields and messages for the `debugging mpls te cspf computation` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beginning pre-process and the next-hop number is ×××.</td>
<td>Preprocessing begins. Number of hops included in the display path is ×××.</td>
</tr>
</tbody>
</table>

Table 11 describes output fields and messages for `debugging mpls te cspf errors` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Others address has been configured to include hop on this node.</td>
<td>Other addresses on this node have been configured in the include hop.</td>
</tr>
</tbody>
</table>

Examples

# Configure MPLS basic capability and MPLS TE basic capability on two directly-connected devices. Execute the `debugging mpls te cspf computation` command. Configure CSPF.

<Sysname> debugging mpls te cspf computation
*Sep 22 02:20:19:989 2003 Sysname CSPF/7/COMPUTE: 01:7252: Begining CSPF computation. The LspId: 9, IngressLsrId: b0b0b0b, EgressLsrId: 38010101.

// CSPF computation started. The LSP ID is 9, ingress LSR ID is b0b0b0b, and egress LSR ID is 38010101.

*Sep 22 02:20:20:167 2003 Sysname CSPF/7/COMPUTE: 01:7262: The TE-class number is 7.

// The TE class number is 7.
*Sep 22 02:20:20:267 2003 Sysname CSPF/7/COMPUTE:
The resource reserve policy is LSPM_RANDOM_FILL.

The first segment computation started.

The current IGP protocol is OSPF, and the process ID is 1.

The pre-process started. The number of hops in the display path is 0.

The start hop in the current segment is 11.11.11.11.

The current end hop is the same as egress ID 56.1.1.1.

The current outgoing flag is 0, and the incoming flag is also 0.

The current computation is loose computation.

The node with router ID 56.1.1.1 is selected.

The current segment’s computation is successful, and the final result is: 0: 87.1.1.1 1: 87.1.1.2

The first segment’s computation is successful.

The CSPF path computation is successful.

The total cost is 1.
The computation of path to egress is finished.

// Computation of the path to the egress has been finished.

# Configure MPLS basic capability and MPLS TE basic capability on two directly-connected devices. Execute the `debugging mpls te cspf events` command. Establish OSPF neighbor relationship between the two devices, and configure CSPF.

<Sysname> debugging mpls te cspf events

*Sep 22 02:31:58:656 2003 Sysname CSPF/7/EVENT: 05:0600: Received 87.1.1.1 link event: ADD.

// CSPF received an add event of link 87.1.1.1.

*Sep 22 02:31:58:765 2003 Sysname CSPF/7/EVENT: 05:0600: Received 87.1.1.2 link event: DELETE.

// CSPF received a delete event of link 87.1.1.2.

*Sep 22 02:31:58:885 2003 Sysname CSPF/7/EVENT: 05:0757: Received network LSA event: MODIFY.

// CSPF received a modify event of network LSA.

*Sep 22 02:31:59:105 2003 Sysname CSPF/7/EVENT: 05:0757: Received network LSA event: MODIFY.

// CSPF received a modify event of network LSA.

*Sep 22 02:31:59:215 2003 Sysname CSPF/7/EVENT: 05:0600: Received 87.1.1.2 link event: DELETE.

// CSPF received a delete event of link 87.1.1.2.

*Sep 22 02:31:59:335 2003 Sysname CSPF/7/EVENT: 05:0757: Received network LSA event: MODIFY.

// CSPF received a network LSA modify event.

*Sep 22 02:31:59:445 2003 Sysname CSPF/7/EVENT: 05:0600: Received 87.1.1.2 link event: ADD.

// CSPF received an add event of link 87.1.1.2.

*Sep 22 02:31:59:555 2003 Sysname CSPF/7/EVENT: 05:0757: Received network LSA event: MODIFY.

// CSPF received a network LSA modify event.

*Sep 22 02:32:00:284 2003 Sysname CSPF/7/EVENT: 05:0757: Received network LSA event: DELETE.

// CSPF received a network LSA delete event.

# Configure MPLS basic capability and MPLS TE basic capability on two directly-connected devices. Execute the `debugging mpls te cspf tedb` command. Establish OSPF neighbor relationship between the two devices, and configure CSPF.

<Sysname> debugging mpls te cspf tedb

*Sep 22 02:37:58:655 2003 Sysname CSPF/7/TEDB: 02:0170: Router node index added successfully to node hash table.

// CSPF added the Router node index to the node hash table.
SEP 22 02:37:58:794 2003 Sysname CSPF/7/TEDB:
02:1872: Router Node for 11.11.11.11 has been added successfully.

// CSPF added the router node 11.11.11.11.

SEP 22 02:37:58:934 2003 Sysname CSPF/7/TEDB:
02:1042: Area Id 0 added successfully to area id list of router node.

// CSPF added area ID 0 to the area ID list of the router node.

SEP 22 02:37:59:74 2003 Sysname CSPF/7/TEDB:
02:1089: Link Index added successfully to Link Index list of router node.

// CSPF added the link Index to the Link Index list of the router node.

SEP 22 02:37:59:224 2003 Sysname CSPF/7/TEDB:
02:0170: Router node index added successfully to node hash table.

// CSPF added the router node index to the node's hash table.

SEP 22 02:37:59:494 2003 Sysname CSPF/7/TEDB:
02:3701: IGP received link info has been processed successfully.

// CSPF processed the IGP link information successfully.

SEP 22 02:37:59:634 2003 Sysname CSPF/7/TEDB:
02:0273: Hash index deleted successfully from node hash table.

// CSPF deleted the hash index from node hash table.

SEP 22 02:37:59:774 2003 Sysname CSPF/7/TEDB:
02:1138: Area Id 0 deleted successfully from area id list of router node.

// CSPF deleted area ID 0 from the area ID list of the router node.

SEP 22 02:37:59:924 2003 Sysname CSPF/7/TEDB:
02:1184: Link Index deleted successfully from Link Index list of router node.

// CSPF deleted the link Index from the Link Index list of the router node.

SEP 22 02:38:00:94 2003 Sysname CSPF/7/TEDB:
02:3040: 87.1.1.2 link has been deleted successfully.

// CSPF deleted link 87.1.1.2.

SEP 22 02:38:00:214 2003 Sysname CSPF/7/TEDB:
02:3701: IGP received link info has been processed successfully.

// CSPF processed the IGP link information successfully.

SEP 22 02:38:00:354 2003 Sysname CSPF/7/TEDB:
02:3284: Network LSA for 87.1.1.2 has been modified successfully.

// CSPF modified the network LSA for 87.1.1.2.

SEP 22 02:38:00:494 2003 Sysname CSPF/7/TEDB:
02:3799: IGP received network LSA has been processed successfully.

// CSPF processed the IGP network LSA successfully.

SEP 22 02:38:00:634 2003 Sysname CSPF/7/TEDB:
02:1089: Link Index added successfully to Link Index list of router node.

// CSPF added the link Index to the Link Index list of the router node.

SEP 22 02:38:00:784 2003 Sysname CSPF/7/TEDB:
02:0170: Router node index added successfully to node hash table.

// CSPF added the router node index to the node’s hash table.

*Sep 22 02:38:00:924 2003 Sysname CSPF/7/TEDB:
02:2749: 87.1.1.2 has been added successfully to link table.

// CSPF added link 87.1.1.2 to the link table.

*Sep 22 02:38:01:55 2003 Sysname CSPF/7/TEDB:
02:3701: IGP received link info has been processed successfully.

// CSPF processed the IGP link information successfully.

*Sep 22 02:38:01:194 2003 Sysname CSPF/7/TEDB:
02:3284: Network LSA for 87.1.1.2 has been modified successfully.

// CSPF modified the network LSA for 87.1.1.2.

*Sep 22 02:38:01:334 2003 Sysname CSPF/7/TEDB:
02:3799: IGP received network LSA has been processed successfully.

// CSPF processed the IGP network LSA successfully.

*Sep 22 02:38:01:484 2003 Sysname CSPF/7/TEDB:
02:0273: Hash index deleted successfully from node hash table.

// CSPF deleted the hash index from the node’s hash table.

*Sep 22 02:38:01:624 2003 Sysname CSPF/7/TEDB:
02:1138: Area Id 0 deleted successfully to area id list of router node.

// CSPF deleted area ID 0 from the area ID list of the router node.

*Sep 22 02:38:01:774 2003 Sysname CSPF/7/TEDB:
02:1184: Link Index deleted successfully from Link Index list of router node.

// CSPF deleted the link index from the Link Index list of the router node.

*Sep 22 02:38:01:934 2003 Sysname CSPF/7/TEDB:
02:3040: 87.1.1.2 link has been deleted successfully.

// CSPF deleted link 87.1.1.2.

*Sep 22 02:38:02:55 2003 Sysname CSPF/7/TEDB:
02:3701: IGP received link info has been processed successfully.

// CSPF processed the IGP link information successfully.

*Sep 22 02:38:02:194 2003 Sysname CSPF/7/TEDB:
02:3284: Network LSA for 87.1.1.2 has been modified successfully.

// CSPF modified the network LSA for 87.1.1.2.

*Sep 22 02:38:02:334 2003 Sysname CSPF/7/TEDB:
02:3799: IGP received network LSA has been processed successfully.

// CSPF processed the IGP network LSA.

*Sep 22 02:38:02:474 2003 Sysname CSPF/7/TEDB:
02:1089: Link Index added successfully to Link Index list of router node.

// CSPF added the link index to the Link Index list of the router node.

*Sep 22 02:38:02:624 2003 Sysname CSPF/7/TEDB:
02:0170: Router node index added successfully to node hash table.

// CSPF added the router node index to the node’s hash table.

*Sep 22 02:38:02:764 2003 Sysname CSPF/7/TEDB:
02:2749: 87.1.1.2 has been added successfully to link table.

74
debugging mpls te management

Use `debugging mpls te management` to enable debugging for MPLS TE management.

Use `undo debugging mpls te management` to disable debugging for MPLS TE management.

Syntax

```
debugging mpls te management { all | auto-bandwidth-adjustment | events | fast-reroute | link-administration | reoptimization | states }
undo debugging mpls te management { all | auto-bandwidth-adjustment | events | fast-reroute | link-administration | reoptimization | states }
```

Default

Debugging for MPLS TE management is disabled.

Views

User view
Default command level

1: Monitor level

Parameters

all: Specifies all types of debugging for MPLS TE management.

auto-bandwidth-adjustment: Specifies debugging for auto-bandwidth-adjustment of MPLS TE management.

events: Specifies debugging for MPLS TE management events.

fast-reroute: Specifies debugging for fast reroute of MPLS TE management.

link-administration: Specifies debugging for link administration of MPLS TE management.

reoptimization: Specifies debugging for reoptimization of MPLS TE management.

states: Specifies debugging for MPLS TE management states.

Examples

The following examples are based on the scenario in which two directly-connected devices have been enabled with MPLS basic capability, MPLS TE basic capability, and MPLS RESV-TE capability.

# Execute the debugging mpls te management auto-bandwidth-adjustment command. Then the two devices establish an RSVP-TE tunnel.

<Sysname> debugging mpls te management auto-bandwidth-adjustment

// MPLS TE management received a global automatic bandwidth adjustment enable event.
*Nov 13 14:20:04:832 2006 Sysname LSPM/7/LSPM TE AUTOBW:
Received Global AutoBw enable event.

// MPLS TE management created an automatic bandwidth adjustment timer for Tunnel 0.
*Nov 13 14:20:04:832 2006 Sysname LSPM/7/LSPM TE AUTOBW:
AutoBw timer created for Tunnel : Tunnel0.

// Automatic bandwidth adjustment timer for Tunnel 0 has expired.
*Nov 13 14:20:05:740 2006 Sysname LSPM/7/LSPM TE AUTOBW:
AutoBw timer expired for Tunnel : Tunnel0.

// LSP agent has been triggered to collect samples for Tunnel 0.
*Nov 13 14:20:05:740 2006 Sysname LSPM/7/LSPM TE AUTOBW:
Lsp Agt triggered to collect sample for Tunnel Tunnel0.

// Sample has been collected for Tunnel 0.

# Execute the debugging mpls te management events command. Then the two devices establish an RSVP-TE tunnel.

<Sysname> debugging mpls te management events

// MPLS TE management received a tunnel create event.
*Nov 13 14:24:56:141 2006 Sysname LSPM/7/LSPM TE EVENTS:
Received Tunnel create event.

// MPLS TE management received the path calculated by CSPF for Tunnel 0.
Main Crlsp is Up for Tunnel : Tunnel0.

// The main CR-LSP for Tunnel 0 is up.

# Execute the `debugging mpls te management fast-reroute` command. Then the two devices establish an RSVP-TE tunnel.

```<Sysname> debugging mpls te management fast-reroute```

The interface has been added for bypass tunnel (Tunnel1) to protect.

// The interface has been successfully added and is to be protected by the bypass tunnel.

# Execute the `debugging mpls te management reoptimization` command. Then the two devices establish an RSVP-TE tunnel.

```<Sysname> debugging mpls te management reoptimization```

MPLS TE management created a reoptimization timer for Tunnel 1.

# Execute the `debugging mpls te management states` command. Then the two devices establish an RSVP-TE tunnel.

```<Sysname> debugging mpls te management states```
Nov 13 14:42:35:130 2006 Sysname LSPM/7/LSPM TE STATES:
State Change :Tunnel Tunnel1 enters READY state from SETUPING state.

// State change: Tunnel 1 entered the READY state from the SETUPING state.

debugging mpls te protect-switch

Use **debugging mpls te protect-switch** to enable debugging for protection switching.
Use **undo debugging mpls te protect-switch** to disable debugging for protection switching.

**Syntax**

debugging mpls te protect-switch { all | error | process | timer }
undo debugging mpls te protect-switch { all | error | process | timer }

**Default**

No protection switching debugging is enabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

**all**: Specifies all types of debugging.
**error**: Specifies debugging for errors.
**process**: Specifies debugging for processing.
**timer**: Specifies debugging for timers.

**Examples**

# Configure MPLS TE tunnels between two routers. Execute the **debugging mpls te protect-switch all** command on the ingress router of the tunnel, and then configure a protection group. The ingress router outputs the following debugging information:

```
<PE1> debugging mpls te protect-switch all
*Feb 22 16:52:53:953 2011 PE1 PSC/7/PROC: Create a PG in PSC: Instance 0, ProtectType 1, ResvertiveModeFlag 1,SwitchMode 1, HoldOff 0, WTR 120.
// Protection switching started to create a protection group.
*Feb 22 16:52:53:953 2011 PE1 PSC/7/PROC: Real-time backup: PG 1, WTnlId 4294967295, PTnlId 4294967295, Action 0.
// Protection switching performed real-time backup for the protection group.
*Feb 22 16:52:53:953 2011 PE1 PSC/7/PROC: Add a path to PSC: Instance 0, PG 1, PathType 0, PContext 79101954.
// Protection switching added a path to the protection group.
*Feb 22 16:52:53:969 2011 PE1 PSC/7/PROC: input event : PSC_LOCAL_EVENT_CSFW
--------before processing----------
    Instance: 0
    PG ID: 1
    State: UNAVAILABLE
```
Old event:
Working path flag: 1
Protection path flag: 1
Switch result: 0
Message: 0x0(0,0)
--------before processing----------
// The protection group was enabled to process the event. This is the protection group's status before processing.
*Feb 22 16:52:53:969 2011 PE1 PSC/7/PROC: Real-time back up: PG 1, WTnlId 79101954, PTnlId 4294967295, Action 1.
// Status changed. The device performed real-time backup for the protection group.
*Feb 22 16:52:53:985 2011 PE1 PSC/7/PROC:
--------after processing----------
Instance: 0
PG ID: 1
State: UNAVAILABLE

Old event:
Working path flag: 0
Protection path flag: 1
Switch result: 0
Message: 0x0(0,0)
--------after processing----------

debugging ospf mpls-te

Use `debugging ospf mpls-te` to enable OSPF TE debugging.
Use `undo debugging ospf mpls-te` to disable OSPF TE debugging.

Syntax

```
debugging ospf [ process-id ] mpls-te
undo debugging ospf [ process-id ] mpls-te
```

Default

OSPF TE debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

`process-id`: Specifies an OSPF process by its ID in the range of 1 to 65535.

Usage guidelines

OSPF TE debugging displays TE LSA receiving, forming, proliferation, and update events.
If no OSPF process number is specified, TE debugging information for all OSPF processes will be output.
Examples

# Enable OSPF TE debugging.
<Sysname> debugging ospf mpls-te
MSDP debugging commands

Support for VPN instances depends on your device model.

The output description tables in this document only contain fields and messages that require an explanation.

debugging msdp

Use debugging msdp to enable MSDP debugging.

Use undo debugging msdp to disable MSDP debugging.

Syntax

debugging msdp [ all-instance | vpn-instance vpn-instance-name ] { all | connect | event | packet | source-active }

undo debugging msdp [ all-instance | vpn-instance vpn-instance-name ] { all | connect | event | packet | source-active }

Default

MSDP debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all-instance: Specifies all MPLS L3VPN instances.

vpn-instance vpn-instance-name: Specifies an MPLS L3VPN instance. The VPN instance name must be a case-sensitive string of 1 to 31 characters and must not contain any spaces.

all: Specifies all types of MSDP debugging.

connect: Specifies debugging for MSDP peer connection resets.

event: Specifies MSDP event debugging.

packet: Specifies MSDP packet debugging.

source-active: Specifies debugging for active sources of MSDP.

Usage guidelines

If neither all-instance nor vpn-instance is specified, this command enables MSDP debugging on the public network.

Table 1 describes output fields and messages for the debugging msdp event command.
Table 58 Output from the debugging msdp event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>static RPF peer</td>
<td>RPF neighbor types:</td>
</tr>
<tr>
<td>E-MBGP peer</td>
<td>• Static RPF peer</td>
</tr>
<tr>
<td>I-MBGP peer</td>
<td>• E-MBGP peer</td>
</tr>
<tr>
<td>NOT BGP peer</td>
<td>• I-MBGP peer</td>
</tr>
<tr>
<td>NOT MBGP peer</td>
<td>• NOT BGP peer</td>
</tr>
<tr>
<td></td>
<td>• NOT MBGP peer</td>
</tr>
</tbody>
</table>

Table 59 Output from the debugging msdp packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA-TLV</td>
<td>TLV for different types of messages:</td>
</tr>
<tr>
<td>SA-Request TLV</td>
<td>• SA-TLV</td>
</tr>
<tr>
<td>SA-Response TLV</td>
<td>• SA-Request TLV</td>
</tr>
<tr>
<td>KeepAlive TLV</td>
<td>• SA-Response TLV</td>
</tr>
<tr>
<td>Notification TLV</td>
<td>• KeepAlive TLV</td>
</tr>
<tr>
<td></td>
<td>• Notification TLV</td>
</tr>
</tbody>
</table>

Table 60 Output from the debugging msdp source-active command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only one peer</td>
<td>Conditions for RPF check:</td>
</tr>
<tr>
<td>Peer is original RP</td>
<td>• Only one peer.</td>
</tr>
<tr>
<td>Peer belongs mesh-group</td>
<td>• Peer is original RP.</td>
</tr>
<tr>
<td>Static RPF peer</td>
<td>• Peer belongs mesh-group.</td>
</tr>
<tr>
<td>MSDP Peer is E-MBGP peer</td>
<td>• Static RPF peer.</td>
</tr>
<tr>
<td>Peer’s AS is the next-AS to RP</td>
<td>• MSDP Peer is E-MBGP peer.</td>
</tr>
</tbody>
</table>
| MSDP Peer is I-MBGP peer   | • Peer’s AS is the next-AS to RP.
| Peer is the next-hop to RP | • MSDP Peer is I-MBGP peer.     |
|                            | • Peer is the next-hop to RP.    |

Examples

# Enable debugging for MSDP events on the public network. When MSDP is enabled, output similar to the following example is generated:

```
<Sysname> debugging msdp event
*Aug 25 09:33:13:130 2006 ar2 MSDP/7/EVENT:
(public net):11.11.11.11: TCP listening (H12726)
// The server 11.11.11.11 started TCP listening.
*Aug 25 09:35:45:790 2006 ar2 MSDP/7/EVENT:
(public net):11.11.11.11: Connection accepted (H12850)
```
*Aug 25 09:35:45:790 2006 ar2 MSDP/7/EVENT:
(public net):11.11.11.11: TCP connection established (H12854)

// A TCP connection was established.

*Aug 25 09:35:45:790 2006 ar2 MSDP/7/EVENT:
(public net):11.11.11.11: Sending message to peer: keepalive (H101045)

// The device sent keepalive messages to the MSDP peer.

*Aug 25 09:35:45:790 2006 ar2 MSDP/7/EVENT:
(public net):11.11.11.11: Originating SA message for peer (H10859)

// The device sent SA messages (if any) to the MSDP peer.

# Enable debugging for MSDP messages on the public network. When MSDP is enabled, output similar to the following example is generated:

<Sysname> debugging msdp packet 

*Aug 25 09:39:07:162 2006 ar2 MSDP/7/PACKET:
(public net):11.11.11.11: Sending a 3-bytes message to peer (H17119)

// The device sent a three-byte message to the MSDP peer.

*Aug 25 09:39:07:162 2006 ar2 MSDP/7/PACKET:
(public net):11.11.11.11: Sending to peer success, 3-bytes sent (H17143)

// The device succeeded in sending the message to the MSDP peer.

*Aug 25 09:39:07:162 2006 ar2 MSDP/7/PACKET:
(public net):11.11.11.11: Received 3-bytes message 1 from peer (H13471)

// The MSDP peer received the three-byte message.

*Aug 25 09:39:07:162 2006 ar2 MSDP/7/PACKET:
(public net):11.11.11.11: KeepAlive TLV (H131441)

// This message is a keepalive message.

# Enable debugging for active sources of MSDP on the public network. When MSDP is enabled, output similar to the following example is generated:

<Sysname> debugging msdp source-active 

*Aug 25 09:52:08:924 2006 ar2 MSDP/7/SOURCE-ACTIVE:
(public net):11.11.11.11: Only one peer, passed RPF check (H132426)

// The SA message was received. The RPF check succeeded because only one MSDP peer exists.
MTR debugging commands

Enabling any debugging command in this chapter might affect system performance, especially when the system is busy. Disable debugging after the debugging operation is complete.

debugging multiple-topology event

Use `debugging multiple-topology event` to enable topology event debugging.
Use `undo debugging multiple-topology event` to disable topology event debugging.

Syntax

```
debugging multiple-topology event
undo debugging multiple-topology event
```

Default

Topology event debugging is disabled.

Views

User view

Default command level

1: Monitor level

Examples

```
# Enable topology event debugging. When you create topology voice, output similar to the following example is generated:
<Sysname> debugging multiple-topology event
<Sysname-address-family ipv4] multiple-topology voice
*Jul 20 11:44:17:282 2009 Sysname MT/7/MTDBG: Create topol:voice
```

debugging multiple-topology hsb

Use `debugging multiple-topology hsb` to enable topology hot backup debugging.
Use `undo debugging multiple-topology hsb` to disable topology hot backup debugging.

Syntax

```
debugging multiple-topology hsb
undo debugging multiple-topology hsb
```

Default

Topology hot backup debugging is disabled.

Views

User view

Default command level

1: Monitor level
# Enable topology hot backup debugging. When you create topology voice, output similar to the following example is generated:

```
<Sysname> debugging multiple-topology hsb
[Sysname-address-family ipv4] multiple-topology voice
// MTR sent information with a backup type of create to the standby card.
```
Multicast routing and forwarding debugging commands

Support for VPN instances depends on your device model.

The output description tables in this document only contain fields and messages that require an explanation.

Some information in this chapter is device type specific. Devices in this chapter are categorized depending on their IRF capability and support for interface cards that use independent processors for forwarding traffic, as shown in Table 1.

**Table 61 Device types**

<table>
<thead>
<tr>
<th>Device type</th>
<th>Interface cards with on-card processors</th>
<th>IRF capability</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed devices</td>
<td>Yes</td>
<td>No</td>
<td>HP 6600 routers (except for 6602)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes (in standalone mode)</td>
<td>HP 12500 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 10500 switches</td>
</tr>
<tr>
<td>Distributed IRF devices</td>
<td>Yes</td>
<td>Yes (in IRF mode)</td>
<td>HP 12500 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 10500 switches</td>
</tr>
<tr>
<td>Centralized devices</td>
<td>No</td>
<td>No</td>
<td>HP MSR routers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 6602 router</td>
</tr>
<tr>
<td>Centralized IRF devices</td>
<td>No</td>
<td>Yes</td>
<td>HP 5800 switches</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HP 5500 switches</td>
</tr>
</tbody>
</table>

debugging mfib

Use `debugging mfib` to enable multicast forwarding information base (MFIB) debugging.

Use `undo debugging mfib` to disable MFIB debugging.

**Syntax**

Centralized devices:

```
debugging mfib [ all-instance | vpn-instance vpn-instance-name ] { all | { driver | no-cache | packet | register | route | sync | upcall | wrong-iif } [ advanced-acl-number ] }
```

```
undo debugging mfib [ all-instance | vpn-instance vpn-instance-name ] { all | { driver | no-cache | packet | register | route | sync | upcall | wrong-iif } [ advanced-acl-number ] }
```

Distributed devices/centralized IRF devices:

```
debugging mfib [ all-instance | vpn-instance vpn-instance-name ] { all | { driver | no-cache | packet | register | route | sync | upcall | wrong-iif } [ advanced-acl-number ] } [ slot slot-number ]
```

86
undo debugging mfib [ all-instance | vpn-instance vpn-instance-name ] { all | { driver | no-cache | packet | register | route | sync | upcall | wrong-iif } [ advanced-acl-number ] } [ slot slot-number ]

Distributed IRF devices:

devices
undo debugging mfib [ all-instance | vpn-instance vpn-instance-name ] { all | { driver | no-cache | packet | register | route | sync | upcall | wrong-iif } [ advanced-acl-number ] } [ chassis chassis-number slot slot-number ]

Default

MFIB debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

**all-instance**: Specifies all the MPLS L3VPN instances.

**vpn-instance vpn-instance-name**: Specifies an MPLS L3VPN instance by its name. The VPN instance name must be a case-sensitive string of 1 to 31 characters and must not contain any spaces.

**all**: Specifies all types of debugging for MFIB.

**driver**: Specifies MFIB debugging for interface drivers.

**no-cache**: Specifies MFIB debugging for unknown multicast packets.

**packet**: Specifies MFIB packet debugging.

**register**: Specifies MFIB debugging for register messages.

**route**: Specifies MFIB route debugging.

**sync**: Specifies MFIB debugging for synchronization messages.

**upcall**: Specifies debugging for messages that MFIB reported to MRM.

**wrong-iif**: Specifies MFIB debugging for incoming interface (IIF) errors.

**advanced-acl-number**: Specifies an advanced ACL number in the range of 3000 to 3999.

**slot slot-number**: Specifies a card by its slot number. (Distributed devices.)

**chassis chassis-number slot slot-number**: Specifies a card on an IRF member device. The chassis-number argument specifies the IRF member device ID. The slot-number argument specifies the number of the slot where the card resides. (Distributed IRF devices.)

Usage guidelines

If neither **all-instance** nor **vpn-instance** is specified, this command enables MFIB debugging for the public network.

Table 2 describes output fields and messages for the debugging mfib driver command.
Table 62 Output from the debugging mfib driver command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>call driver</td>
<td>The MFIB called the driver.</td>
</tr>
<tr>
<td>Do not add to driver</td>
<td>The MFIB did not add the multicast forwarding entries to the driver.</td>
</tr>
<tr>
<td>downloaded to driver Failed</td>
<td>The MFIB failed to issue configurations to the driver.</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>set unknown packet to CPU</td>
<td>The MFIB sent unknown packets to the CPU for process.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the **debugging mfib no-cache** command.

Table 63 Output from the debugging mfib no-cache command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive no cache report</td>
<td>The MFIB received an unknown multicast data packet.</td>
</tr>
<tr>
<td>No MFIB entry matches</td>
<td>The MFIB failed to find the matching entry.</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>Cache the packet</td>
<td>The MFIB cached multicast data packets.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the **debugging mfib packet** command.

Table 64 Output from the debugging mfib packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive/Drop packet</td>
<td>The MFIB received or dropped multicast data packets.</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>IIF</td>
<td>Incoming interface.</td>
</tr>
<tr>
<td>OIF</td>
<td>Outgoing interface.</td>
</tr>
<tr>
<td>TTL</td>
<td>TTL value of the message.</td>
</tr>
<tr>
<td>InPktVrf</td>
<td>Index number of the VPN to which the received packet belongs.</td>
</tr>
<tr>
<td>Forward multicast packet</td>
<td>The MFIB forwarded multicast data packets.</td>
</tr>
<tr>
<td>DP</td>
<td>Data plane (supported only by multi-core devices).</td>
</tr>
<tr>
<td>control plane</td>
<td>Control plane (supported only by multi-core devices).</td>
</tr>
<tr>
<td>Send packet to control plane</td>
<td>The data plane sent multicast packets to the control plane (supported only by multi-core devices).</td>
</tr>
<tr>
<td>Drop packet because unknown packet to cpu function is closed</td>
<td>The MFIB dropped unknown packets because the function for sending unknown packets to CPU was disabled (supported only by multi-core devices).</td>
</tr>
<tr>
<td>Search entry in DP MFIB</td>
<td>The MFIB searched the data plane MFIB for correct entries (supported only by multi-core devices).</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the **debugging mfib register** command.
### Table 65 Output from the debugging mfib register command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send register</td>
<td>The MFIB sent a register message.</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>Dropping received</td>
<td>The MFIB dropped error register messages.</td>
</tr>
<tr>
<td>register packet</td>
<td></td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the **debugging mfib route** command.

### Table 66 Output from the debugging mfib route command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add-entry message</td>
<td>The MFIB received an add-entry message from the multicast routing module (MRM).</td>
</tr>
<tr>
<td>del-entry message</td>
<td>The MFIB received a del-entry message from MRM.</td>
</tr>
<tr>
<td>set-IIF message</td>
<td>The MFIB received a set-IIF message from MRM.</td>
</tr>
<tr>
<td>del-OIF message</td>
<td>The MFIB received a del-OIF message from MRM.</td>
</tr>
<tr>
<td>add-OIF message</td>
<td>The MFIB received an add-OIF message from MRM.</td>
</tr>
<tr>
<td>MRM</td>
<td>Multicast routing module.</td>
</tr>
<tr>
<td>The Following OIFs are added</td>
<td>The following outgoing interfaces were added.</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>Add/Delete entry successfully</td>
<td>The data plane successfully added or deleted an entry (supported only by multi-core devices).</td>
</tr>
<tr>
<td>Add/Delete num OIF(s)</td>
<td>The data plane entry successfully added or deleted outgoing interfaces, the number of which is specified by num (supported only by multi-core devices).</td>
</tr>
<tr>
<td>Set Switch-Group successfully</td>
<td>The data plane entry successfully configured switch-group (supported only by multi-core devices).</td>
</tr>
</tbody>
</table>

Table 7 describes output fields and messages for the **debugging mfib sync** command.

### Table 67 Output from the debugging mfib sync command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>added to updated list</td>
<td>The MFIB added an entry to the updated list.</td>
</tr>
<tr>
<td>deleted from updated list</td>
<td>The MFIB deleted an entry from the updated list.</td>
</tr>
<tr>
<td>Encoded the ADD/DEL/MOD message</td>
<td>The MFIB encapsulated a message to add/delete/modify an entry.</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
</tbody>
</table>

Table 8 describes output fields and messages for the **debugging mfib upcall** command.

### Table 68 Output from the debugging mfib upcall command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send No cache up call</td>
<td>The MFIB sent a message to MRM.</td>
</tr>
</tbody>
</table>
Table 9 describes output fields and messages for the debugging mfib wrong-iif command.

### Table 69 Output from the debugging mfib wrong-iif command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>WRONG_IF packet</td>
<td>The MFIB received a multicast data packet on a wrong incoming interface.</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
</tbody>
</table>

**Examples**

- **On multi-core devices:**

  # Enable MFIB packet debugging for the public network. When PIM-DM is enabled on interfaces, output similar to the following example is generated:

  ```
  <Sysname> debugging mfib packet
  *Nov 12 17:25:58:202 2007 Sysname DP_MFIB/7/PACKET:
  (IPv4 0): (3.4.5.6, 226.1.1.1) Receive packet from interface Ethernet1/1 in vcpu 7. (D45208)
  // The MFIB received a multicast packet (3.4.5.6, 226.1.1.1) from Ethernet 1/1 on VCPU 7.
  *Nov 12 17:25:58:204 2007 Sysname DP_MFIB/7/PACKET:
  (IPv4 0): (3.4.5.6, 226.1.1.1) Search entry in DP MFIB. (D45263)
  // The MFIB searched the data plane MFIB for the forwarding entry for the multicast packet (3.4.5.6, 226.1.1.1).
  *Nov 12 17:25:58:204 2007 Sysname DP_MFIB/7/PACKET:
  (IPv4 0): (3.4.5.6, 226.1.1.1) Send packet to control plane. (D45314)
  // The data plane sent the multicast packet (3.4.5.6, 226.1.1.1) to the control plane.
  *Nov 12 17:25:58:204 2007 Sysname MFIB/7/MFIB PACKET:
  (public net):Receive packet (3.4.5.6, 226.1.1.1), iif = Ethernet1/1, TTL = 128, InPktVrf = 0(A08238)
  // The MFIB received a public network multicast packet (3.4.5.6, 226.1.1.1) with the TTL value of 128.
  *Nov 12 17:25:58:205 2007 Sysname MFIB/7/MFIB PACKET:
  (public net):No MFIB entry matches (3.4.5.6, 226.1.1.1)(A08351)
  // No matching multicast forwarding entry was found.
  *Nov 12 17:25:58:206 2007 Sysname MFIB/7/MFIB PACKET:
  (public net):Cache the packet for (3.4.5.6, 226.1.1.1)(A082126)
  // The MFIB cached the multicast packet (3.4.5.6, 226.1.1.1).
  *Nov 12 17:25:58:207 2007 Sysname MFIB/7/MFIB PACKET:
  (public net):Forward multicast packet (3.4.5.6, 226.1.1.1) Ethernet1/1 (A081013)
  // The MFIB forwarded the multicast packet (3.4.5.6, 226.1.1.1) to local Ethernet1/1(A084106)
  // The MFIB forwarded the multicast packet (3.4.5.6, 226.1.1.1) out of Ethernet 1/1.
  ```
# Enable MFIB route debugging for the public network. When PIM-DM is enabled on interfaces, output similar to the following example is generated:

```bash
<Sysname> debugging mfib route

*Nov 12 17:43:01:491 2007 Sysname MFIB/7/MFIB ROUTE:
(public net):Receive no cache report,add dummy entry (3.4.5.6, 226.1.1.1)(A07109)
*Nov 12 17:43:01:492 2007 Sysname MFIB/7/MFIB ROUTE:
(public net):Entry (3.4.5.6, 226.1.1.1) is added to dummy list(A063866)
*Nov 12 17:43:01:492 2007 Sysname MFIB/7/MFIB ROUTE:
(public net):The Dummy entry (3.4.5.6, 226.1.1.1) is added(A061861)

// The control plane received a multicast packet (3.4.5.6, 226.1.1.1). No forwarding entry for the packet was found. The control plane created a dummy entry for it.
*Nov 12 17:43:01:492 2007 Sysname DP_MFIB/7/ROUTE:
(IPv4 0): (3.4.5.6, 226.1.1.1) Add entry successfully! (D44157)

// The data plane also created an entry for the multicast packet (3.4.5.6, 226.1.1.1).
*Nov 12 17:43:01:495 2007 Sysname MFIB/7/MFIB ROUTE:
(public net):Receive add-entry message of entry (3.4.5.6, 226.1.1.1) from MRM, OIF num is 1.(A112135)

// An add-entry message was received from the MRM for adding an entry for (3.4.5.6, 226.1.1.1) with the outgoing interface number of 1.
*Nov 12 17:43:01:495 2007 Sysname DP_MFIB/7/ROUTE:
(IPv4 0): (3.4.5.6, 226.1.1.1) Delete entry successfully! (D44246)
*Nov 12 17:43:01:496 2007 Sysname MFIB/7/MFIB ROUTE:
(public net):Entry (3.4.5.6, 226.1.1.1) is deleted from dummy list(A063908)
*Nov 12 17:43:01:497 2007 Sysname MFIB/7/MFIB ROUTE:
(public net):The dummy entry (3.4.5.6, 226.1.1.1) is replaced by actual entry(A07407)
*Nov 12 17:43:01:497 2007 Sysname MFIB/7/MFIB ROUTE:
(public net):The entry (3.4.5.6, 226.1.1.1) is added(A0669)

// The MFIB deleted the dummy entry for (3.4.5.6, 226.1.1.1) from the data plane table and created a forwarding entry for (3.4.5.6, 226.1.1.1).
```
```

- On single-core devices:

  # Enable MFIB driver debugging for the public network. When PIM-DM is enabled on interfaces, output similar to the following example is generated:

  ```bash
  <Sysname> debugging mfib driver
  *Apr 26 12:53:18:967 2000 Sysname MFIB/7/MFIB DRIVER:
  (public net):Succeed to set IPMC 0x0 on interface Vlan-interface20(A082415)
  ```

  // Multicast routing was enabled on VLAN-interface 20.

  ```bash
  *Apr 26 12:53:18:979 2000 Sysname MFIB/7/MFIB DRIVER:
  (public net):Succeed to set unknown packet to CPU 0x3 on interface Vlan-interface20(A082520)
  ```

  // VLAN-interface 20 was configured to send unknown data packets to the CPU.

```

91
A router does not output interface driver debugging information about MFIB.

# Enable MFIB unknown packet debugging for the public network. When PIM-DM is enabled on interfaces, output similar to the following example is generated:

<Sysname> debugging mfib no-cache

*Apr 26 12:43:19:09 2000 Sysname MFIB/7/MFIB NO-CACHE:
(public net):No MFIB entry matches (3.4.5.6, 226.1.1.1)(A08303)

*Apr 26 12:43:19:17 2000 Sysname MFIB/7/MFIB NO-CACHE:
(public net):Receive no cache report,add dummy entry (3.4.5.6, 226.1.1.1)(A07120)

*Apr 26 12:43:19:27 2000 Sysname MFIB/7/MFIB NO-CACHE:
(public net):Cache the packet for (3.4.5.6, 226.1.1.1)(A082099)

//The MFIB received a multicast packet (3.4.5.6, 226.1.1.1). No forwarding entry was found for the packet. The MFIB created a dummy forwarding entry and cached the multicast packet.

# Enable MFIB packet debugging for the public network. When PIM-DM is enabled on interfaces, output similar to the following example is generated:

<Sysname> debugging mfib packet

*Apr 26 12:28:50:578 2000 Sysname MFIB/7/MFIB PACKET:
(public net):Receive packet (3.4.5.6, 226.1.1.1), iif = Vlan-interface20, TTL = 128(A08204)

// The MFIB received a multicast packet (3.4.5.6, 226.1.1.1) with a TTL value of 128 on VLAN-interface 20.

*Apr 26 12:28:50:586 2000 Sysname MFIB/7/MFIB PACKET:
(public net):No MFIB entry matches (3.4.5.6, 226.1.1.1)(A08303)

// The MFIB failed to find a matching multicast forwarding entry.

*Apr 26 12:28:50:605 2000 Sysname MFIB/7/MFIB PACKET:
(public net):Cache the packet for (3.4.5.6, 226.1.1.1)(A082099)

// The MFIB cached the multicast packet (3.4.5.6, 226.1.1.1).

*Apr 26 12:28:50:615 2000 Sysname MFIB/7/MFIB PACKET:
(public net):Forward multicast packet (3.4.5.6, 226.1.1.1) on Vlan-interface10(A08945)

// The MFIB forwarded the multicast packet (3.4.5.6, 226.1.1.1) to VLAN-interface 10.

*Apr 26 12:28:50:625 2000 Sysname MFIB/7/MFIB PACKET:
(public net):Forward multicast packet (3.4.5.6, 226.1.1.1) at port Ethernet1/0 (A083551)

// The MFIB forwarded the multicast packet (3.4.5.6, 226.1.1.1) out of Ethernet 1/1.

# Enable MFIB debugging for register messages for the public network. The output in this example was created when the following conditions exist:

- PIM-SM is enabled on the two devices.
- The RP and the BSR are configured.

<Sysname> debugging mfib register
The MFIB received a register message (22.1.1.1, 10.1.1.1) with the encapsulated multicast packet (22.1.1.10, 226.1.1.1).

The MFIB sent a register-stop message for (22.1.1.10, 226.1.1.1) to 22.1.1.1.

Enable MFIB route debugging for the public network. When PIM-DM is enabled on interfaces, output similar to the following example is generated:

Enable MFIB synchronization message debugging for the public network. When PIM-DM is enabled on interfaces, output similar to the following example is generated:

Enable debugging for packets that the MFIB reported to MRM for the public network. When PIM-DM is enabled on interfaces, output similar to the following example is generated:

Enable MFIB debugging for incoming interface errors for the public network. The output in this example was created when the following conditions exist:

93
PIM-DM is enabled on VLAN-interface 40 and VLAN-interface 60.

Packets with the same multicast source address and multicast group address are sent to the two interfaces.

```<Sysname> debugging mfib wrong-iif
*Jan 24 04:36:52:990 2003 Sysname MFIB/7/MFIB WRONG-IIF:
(public net):Slot=3;WRONG_IF packet (10.11.113.168, 226.1.1.1) received on Vlan-interface60, should from Vlan-interface40(A08734)
// The MFIB received a multicast packet (10.11.113.168, 226.1.1.1) on an incorrect incoming interface (VLAN-interface 60). The correct incoming interface is VLAN-interface 40.```

**debugging mrm**

Use `debugging mrm` to enable MRM debugging.

Use `undo debugging mrm` to disable MRM debugging.

**Syntax**

```debugging mrm [ all-instance | vpn-instance vpn-instance-name ] { all | event | packet [ advanced-acl-number ] | route [ advanced-acl-number ] }
undo debugging mrm [ all-instance | vpn-instance vpn-instance-name ] { all | event | packet | route }
```

**Default**

MRM debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- `all-instance`: Specifies all the MPLS L3VPN instances.
- `vpn-instance vpn-instance-name`: Specifies an MPLS L3VPN instance by its name. The VPN instance name must be a case-sensitive string of 1 to 31 characters and must not contain any spaces.
- `all`: Specifies all types of debugging for MRM.
- `event`: Specifies MRM event debugging.
- `packet`: Specifies MRM packet debugging.
- `route`: Specifies MRM route debugging.
- `advanced-acl-number`: Specifies an advanced ACL number in the range of 3000 to 3999.

**Usage guidelines**

If neither `all-instance` nor `vpn-instance` is specified, this command enables MRM debugging for the public network.

Table 10 describes output fields and messages for the `debugging mrm event` command.
Table 70 Output from the debugging mrm event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
<tr>
<td>failed</td>
<td>Operation failed.</td>
</tr>
</tbody>
</table>

Table 11 describes output fields and messages for the debugging mrm packet command.

Table 71 Output from the debugging mrm packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Received</td>
<td>Received packets.</td>
</tr>
<tr>
<td>MFIB information(NOCACHE)</td>
<td>Types of received packets:</td>
</tr>
<tr>
<td>MFIB information(WRONGIF)</td>
<td>• MFIB information(NOCACHE)</td>
</tr>
<tr>
<td>MFIB information(ACTIVE)</td>
<td>• MFIB information(WRONGIF)</td>
</tr>
<tr>
<td>MFIB information(INACTIVE)</td>
<td>• MFIB information(ACTIVE)</td>
</tr>
<tr>
<td>MFIB information(SPT)</td>
<td>• MFIB information(INACTIVE)</td>
</tr>
<tr>
<td>MFIB information(CLEAR)</td>
<td>• MFIB information(SPT)</td>
</tr>
<tr>
<td>MFIB information(REG-Timeout)</td>
<td>• MFIB information(CLEAR)</td>
</tr>
<tr>
<td>Pim packet (protocol = 2)</td>
<td>• Pim</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>• packet (protocol = 2)</td>
</tr>
</tbody>
</table>

(sadd, gadd) (S, G) entry.

Table 12 describes output fields and messages for the debugging mrm route command.

Table 72 Output from the debugging mrm route command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lost the route</td>
<td>A route was deleted.</td>
</tr>
<tr>
<td>a new route</td>
<td>A new route was added.</td>
</tr>
<tr>
<td>(sadd, gadd)</td>
<td>(S, G) entry.</td>
</tr>
</tbody>
</table>

Examples

The output in the following examples was created when PIM-SM is enabled:

# Enable MRM event debugging for the public network.
<Sysname> debugging mrm event
*0.1433697 85 MRM/7/EVENT:
(public net):Task(IGMPV3) register interest in (*,235.1.1.1)(C153622)
// An IGMP join message was received and IGMPv3 created a (*, 235.1.1.1) entry.
*0.1434697 85 MRM/7/EVENT:
(public net):Task(IGMPV3) unregister interest in (*,235.1.1.1)(C153522)
// IGMPv3 deleted the (*, 235.1.1.1) entry 130 seconds after the host stopped sending IGMP join messages.

# Enable MRM packet debugging on the public network.
<Sysname> debugging mrm packet
// The MRM received a multicast packet (1.1.1.108, 235.1.1.1) for an unknown group address from the MFIB on the interface with the index number of 0x30F0188.

// The multicast forwarding entry of (1.1.1.108, 235.1.1.1) transited to INACTIVE state 210 seconds after the multicast source stopped sending data stream.

# Enable MRM route debugging on the public network.
<Sysname> debugging mrm route

// The MRM found a new route for the entry (3.3.3.156, 225.0.0.1).

// The route to 100.1.1.1 is reachable and the record for the entry (3.3.3.156, 225.0.0.1) was moved from the null list.

// The MRM received a creation alert for multicast forwarding entry (3.3.3.156, 225.0.0.1).

**debugging mtracert**

Use **debugging mtracert** to enable multicast traceroute debugging.

Use **undo debugging mtracert** to disable multicast traceroute debugging.

**Syntax**

```
debugging mtracert { all | event | packet }
undo debugging mtracert { all | event | packet }
```

**Default**

Multicast traceroute debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **all**: Specifies all types of multicast traceroute debugging.
- **event**: Specifies multicast traceroute event debugging.
- **packet**: Specifies multicast traceroute packet debugging.
Usage guidelines

Table 13 describes output fields and messages for the `debugging mtracert event` command.

Table 73 Output from the debugging mtracert event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task send buffer</td>
<td>Buffer for sending task.</td>
</tr>
<tr>
<td>Maxhops</td>
<td>Maximum number of hops.</td>
</tr>
<tr>
<td>Rttl</td>
<td>TTL value of the multicast traceroute packet header.</td>
</tr>
<tr>
<td>Multicast-enabled</td>
<td>Enable IP multicast routing.</td>
</tr>
<tr>
<td>Responder</td>
<td>Address of the responder.</td>
</tr>
<tr>
<td>Ip_chPr</td>
<td>Protocol field in the IP packet header.</td>
</tr>
<tr>
<td>Appointed response</td>
<td>Response address designated by the trace command.</td>
</tr>
</tbody>
</table>

Table 13 describes output fields and messages for the `debugging mtracert packet` command.

Table 74 Output from the debugging mtracert packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error hops</td>
<td>Incorrect hop count (the number of hops in the IGMP message header)</td>
</tr>
<tr>
<td>Response block number</td>
<td>Number of response blocks.</td>
</tr>
</tbody>
</table>

Examples

The output in the following examples was created when you use the `mtracert` command to trace the multicast packet path on an IP multicast routing enabled device:

# Enable debugging for multicast traceroute events.

```bash
<Sysname> debugging mtracert event
*Jan 31 17:40:01:522 2007 Sysname MTRACE/7/event: Can't locate upstream neighbor for (10.10.10.8, 225.2.1.1) in multicast routing table (FWD Code: NO_ROUTE)  (M07520)
// The multicast traceroute was unable to locate the upstream neighbor for (10.10.10.8, 225.2.1.1).
```

# Enable debugging for multicast traceroute packets.

```bash
<Sysname> debugging mtracert packet
*Jan 31 17:36:06:221 2007 Sysname MTRACE/7/packet: Receive mtrace response packet from 7.7.7.2 (M12480)
// The multicast traceroute received a response packet from 7.7.7.2.
```
Multicast VPN debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging md

Use **debugging md** to enable multicast domain (MD) debugging.

Use **undo debugging md** to disable MD debugging.

**Syntax**

```
debugging md [ all-instance | vpn-instance vpn-instance-name ] { all | event [ advanced-acl-number ] | packet | timer }
undo debugging md [ all-instance | vpn-instance vpn-instance-name ] { all | event | packet | timer }
```

**Default**

MD debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **all-instance**: Specifies all MPLS L3VPN instances.
- **vpn-instance vpn-instance-name**: Specifies an MPLS L3VPN instance by its name. The VPN instance name must be a case-sensitive string of 1 to 31 characters and must not contain any spaces.
- **all**: Specifies all types of MD debugging.
- **event**: Specifies MD event debugging.
- **advanced-acl-number**: Specifies an advanced ACL number in the range of 3000 to 3999.
- **packet**: Specifies MD packet debugging.
- **timer**: Specifies MD timer debugging.

**Usage guidelines**

If neither **all-instance** nor **vpn-instance** is specified, this command enables MD debugging on the public network.

If a debugging option is enabled for all VPN instances, that debugging option is automatically enabled for VPN instances created afterwards.

Table 1 describes output fields and messages for the **debugging md event** command.
Table 75 Output from the debugging md event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>local MDT info</td>
<td>Local MDT route information.</td>
</tr>
<tr>
<td>sender</td>
<td>Sender of the MDT switching notification.</td>
</tr>
<tr>
<td>source list</td>
<td>List of senders of MDT switching notifications.</td>
</tr>
<tr>
<td>Pending node</td>
<td>Node with a pending join.</td>
</tr>
<tr>
<td>Join Pending Job</td>
<td>Job that processes a pending join.</td>
</tr>
<tr>
<td>switch receive table</td>
<td>Table was created after a switching notification message was received. The table is used for storing switch-group addresses and information about switching notification senders, including the switch group and initiator addresses.</td>
</tr>
<tr>
<td>Join Notification</td>
<td>A join notification.</td>
</tr>
<tr>
<td>SR</td>
<td>Switch receive module.</td>
</tr>
<tr>
<td>Source-MDT</td>
<td>The Source-MDT is the same as the switch-MDT.</td>
</tr>
<tr>
<td>UP/DOWN event</td>
<td>Above-threshold/below-threshold event.</td>
</tr>
<tr>
<td>SS walk</td>
<td>All (S, G) entries in the specified instance are examined in the switch-MDT switching threshold and ACL rule configuration.</td>
</tr>
<tr>
<td>PD</td>
<td>Public MD module.</td>
</tr>
<tr>
<td>SD_QueryShareGroup</td>
<td>Share-group query function.</td>
</tr>
<tr>
<td>PD_JoinGroup</td>
<td>Group join function.</td>
</tr>
<tr>
<td>Resume Source-MDT state</td>
<td>MD reset the switch-MDT state.</td>
</tr>
<tr>
<td>Allocate Switch-Group</td>
<td>MD allocated a switch-group address.</td>
</tr>
<tr>
<td>Recycle Switch-Group</td>
<td>MD recycled the switch-group address. After this operation, traffic will not be forwarded by using the switch-MDT. The switch-group configuration remains the same.</td>
</tr>
<tr>
<td>Release Switch-Group</td>
<td>MD released the switch-group and deleted the switch-group configuration.</td>
</tr>
<tr>
<td>SS_SWITCH_INITIAL</td>
<td>This (S, G) entry has just been created.</td>
</tr>
<tr>
<td>SS_SWITCH_DELAY</td>
<td>This (S, G) entry is in switch-delay state.</td>
</tr>
<tr>
<td>SS_SWITCH_HOLDDOWN</td>
<td>This (S, G) entry is in switch-holddown state.</td>
</tr>
<tr>
<td>SS_SWITCH_NORMAL</td>
<td>This (S, G) entry is in switch-normal state.</td>
</tr>
<tr>
<td>SS_HOLDOWNBYCLEAR</td>
<td>This (S, G) entry is in switch-holddown state because the reset command was executed.</td>
</tr>
<tr>
<td>SS_UNKNOWN</td>
<td>This (S, G) entry is in unknown state.</td>
</tr>
<tr>
<td>Switch-Send timer</td>
<td>Timer for sending MDT switching notification.</td>
</tr>
<tr>
<td>Switch-Delay state</td>
<td>The traffic was being forwarded by using the share-MDT. When the MDT switching conditions were met, the traffic switched to the switch-MDT.</td>
</tr>
<tr>
<td>Switch-Holdown state</td>
<td>The traffic was being forwarded by using the switch-MDT. If the MDT switching conditions were no longer met, the traffic switched back to the share-MDT.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
SwitchReceiverTimer | Timer for receiving MDT switching notification.
IFC_ADD event | Interface adding event.
IFC_UP event | Interface up event.
IFC_DELETE event | Interface deletion event.

Table 2 describes output fields and messages for the `debugging md packet` command.

**Table 76 Output from the debugging md packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch notification message</td>
<td>MDT switching notification message.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the `debugging md timer` command.

**Table 77 Output from the debugging md timer command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query-MTI-Address timer</td>
<td>MTI address query timer.</td>
</tr>
<tr>
<td>BGP Addr</td>
<td>Obtained BGP address. The MTI address is consistent with this address.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable MD event debugging for VPN instance `mvpn`. Output similar to the following example is generated under the following conditions:

- A share-group with the group address of 225.1.2.3 is created.
- MTI0 is associated with the VPN instance `mvpn`.

```bash
<Sysname> debugging md vpn-instance mvpn event
*Jun 16 08:09:24:452 2008 Sysname MD/7/MD:(mvpn): MD Task created successfuly for InstanceID 1(M04790)
*Jun 16 08:09:24:453 2008 Sysname MD/7/MD:(mvpn): IFC_ADD event is received for MTunnel0(M041115)
// An MD task was successfully created and an IFC_ADD event for MTI interface 0 was received.
*Jun 16 08:09:24:472 2008 Sysname MD/7/MD:(mvpn): Share-Group 225.1.2.3 is added successfully for Instance mvpn(M04177)
*Jun 16 08:09:24:472 2008 Sysname MD/7/MD:(mvpn): Notifying MD enable for Instance 1 to BGP(M04253)
*Jun 16 08:09:24:512 2008 Sysname MD/7/MD:(mvpn): Interface MTunnel0 is added successfuly in Instance mvpn(M041194)
// MD successfully created the share-group in VPN instance mvpn and bound the MTI to the VPN instance. The VPN instance sent a notification that MD was enabled in the VPN instance to BGP.
*Jun 16 08:09:24:512 2008 Sysname MD/7/MD:(mvpn): IFC_ADD event is received for MTunnel0 for Instance 1(M041261)
*Jun 16 08:09:24:573 2008 Sysname MD/7/MD:(mvpn): IFC_UP event is received for MTunnel0 for Instance 1(M041299)
*Jun 16 08:09:24:574 2008 Sysname MD/7/MD:(mvpn): Notifying add local MDT info (225.1.2.3, 12.34.56.78) for Instance 1 to BGP(M041319)
```
// After MTI got up, an up event was received. BGP was informed to add local MDT routing information.
*Jun 16 08:21:50:706 2008 Sysname MD/7/MD:(mvpn): Send Join Group 225.1.2.3 to PD for InstanceID 1 (M04882)
*Jun 16 08:21:50:706 2008 Sysname MD/7/MD:(mvpn): Socket initialization is successful for MTunnel0 (12.34.56.78) (M04661)
*Jun 16 08:21:50:707 2008 Sysname MD/7/MD:(mvpn): PIM Initialization on MTI is done successfully for Instance 1 (M04922)

// PIM-enabled interfaces exist in the VPN instance. The VPN instance sent a join notification to the public network for joining the share-group, and the same PIM mode was initialized on all the MTI interfaces.

# Enable MD packet debugging for VPN instance mvpn. When a remote PE device initiates an MDT switchover, output similar to the following example is generated:
<Sysname> debugging md vpn-instance mvpn packet
*Mar 12 17:28:21:709 2007 Sysname MD/7/MD:(mvpn): Number of MD TLV received is 1 (M03197)
*Mar 12 17:28:21:953 2007 Sysname MD/7/MD:(mvpn): Refresh Time for 226.4.5.0 switch group & sender 22.22.22.22 (M03562)

// MD received a MDT switching notification message from a remote PE. The locally attached PE received the packet (10.21.21.3, 235.4.5.6), and the local device refreshed the switching receiving timer.

# Enable MD timer debugging for VPN instance mvpn. When the MTI address query timer expires, output similar to the following example is generated:
<Sysname> debugging md vpn-instance mvpn timer
*Mar 12 17:29:50:428 2007 Sysname MD/7/MD: Query-MTI-Address timer expires (M042864)
*Mar 12 17:29:50:524 2007 Sysname MD/7/MD: Result: 0, BGP Addr: 11.11.11.11 (M042881)
*Mar 12 17:29:55:524 2007 Sysname MD/7/MD: Result: 0, BGP Addr: 11.11.11.11 (M042881)

// When the MTI address query timer (set to 5 seconds) expired, the MTI address was automatically refreshed to keep consistent with the obtained BGP address.

debugging mtunnel

Use debugging mtunnel to enable multicast tunnel debugging.
Use undo debugging mtunnel to disable multicast tunnel debugging.

Syntax
debugging mtunnel { all | error | event | ipc | management | packet }
undo debugging mtunnel { all | error | event | ipc | management | packet }

Default
Multicast tunnel debugging is disabled.

Views
User view
Default command level

1: Monitor level

Parameters

all: Specifies all types of multicast tunnel debugging.
error: Specifies multicast tunnel error debugging.
event: Specifies multicast tunnel event debugging.
ipc: Specifies multicast tunnel IPC message debugging.
management: Specifies multicast tunnel management debugging.
packet: Specifies multicast tunnel packet debugging.

Usage guidelines

Table 4 describes output fields and messages for the debugging mtunnel error command.

Table 78 Output from the debugging mtunnel error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The interface index is invalid</td>
<td>The MTI index is invalid.</td>
</tr>
<tr>
<td>Get-LinkCB is failed</td>
<td>The multicast tunnel failed to obtain the MT link-layer control block.</td>
</tr>
<tr>
<td>pstMTunnel is NULL</td>
<td>The link-layer control block pointer obtained through the MTI index is null.</td>
</tr>
<tr>
<td>pData is NULL</td>
<td>The function input parameter pointer is null.</td>
</tr>
<tr>
<td>pPhylInfo is NULL</td>
<td>The physical statistics memory pointer of MT is null.</td>
</tr>
<tr>
<td>The PhyInfo is too large</td>
<td>The amount of statistics data to be output exceeded the buffer size.</td>
</tr>
<tr>
<td>The parameter is NULL</td>
<td>The set interface parameter pointer is null.</td>
</tr>
<tr>
<td>The pulDrvContext is NULL</td>
<td>The memory address for obtaining the DrvContext value is null.</td>
</tr>
<tr>
<td>prepare Drv-block is failed</td>
<td>The multicast tunnel failed to request memory for driver data.</td>
</tr>
<tr>
<td>send to DRV is failed</td>
<td>The multicast tunnel failed to deliver driver.</td>
</tr>
<tr>
<td>MTunnel_ReceivePacket is NULL</td>
<td>Packet pointer received by the MT is null.</td>
</tr>
<tr>
<td>Preparing MTunnelCB is failed</td>
<td>The multicast tunnel failed to request memory for MTI link-layer data blocks.</td>
</tr>
<tr>
<td>Preparing PhyInfo-block is failed</td>
<td>The multicast tunnel failed to request memory for MTI physical statistics data blocks.</td>
</tr>
<tr>
<td>Get default Mtunnel mode is failed</td>
<td>The multicast tunnel failed to obtain the MT default mode.</td>
</tr>
<tr>
<td>Init MtunnelCB is failed</td>
<td>The multicast tunnel failed to initialize the MT link-layer data block.</td>
</tr>
<tr>
<td>The packet is NULL</td>
<td>The packet pointer is null.</td>
</tr>
<tr>
<td>Can't get MTunnel PhyInfo Ptr</td>
<td>The multicast tunnel failed to obtain the MT physical statistics data block.</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the debugging mtunnel event command.
### Table 79 Output from the debugging mtunnel event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The block chain is NULL</td>
<td>The global chain of MT data block is null.</td>
</tr>
<tr>
<td>SplImp is failed</td>
<td>The multicast tunnel failed to disable interruption.</td>
</tr>
<tr>
<td>the DrvContext[0] is 0x</td>
<td>DrvContext[0] value of the MPU.</td>
</tr>
<tr>
<td>the DrvContext[1] is 0x</td>
<td>DrvContext[1] value of the MPU.</td>
</tr>
<tr>
<td>the DrvContext[0] from DRV is 0x</td>
<td>DrvContext[0] value the MPU obtained from the driver.</td>
</tr>
<tr>
<td>the DrvContext[1] from DRV is 0x</td>
<td>DrvContext[1] value the MPU obtained from the driver.</td>
</tr>
<tr>
<td>the IO-DrvContext[0] from mainboard is 0x</td>
<td>DrvContext[0] value the IO card obtained from the MPU.</td>
</tr>
<tr>
<td>the IO-DrvContext[1] from mainboard is 0x</td>
<td>DrvContext[1] value the IO card obtained from the MPU.</td>
</tr>
<tr>
<td>the IO-DrvContext[0] from IOboard is 0x</td>
<td>DrvContext[0] value the IO card.</td>
</tr>
<tr>
<td>the IO-DrvContext[1] from IOboard is 0x</td>
<td>DrvContext[1] value the IO card.</td>
</tr>
<tr>
<td>the IO-DrvContext[0] from DRV is 0x</td>
<td>DrvContext[0] value the IO card obtained from the driver.</td>
</tr>
<tr>
<td>the IO-DrvContext[1] from DRV is 0x</td>
<td>DrvContext[1] value the IO card obtained from the driver.</td>
</tr>
<tr>
<td>Deleted link CB(if index = 0x%X)</td>
<td>The MT link-layer control block was successfully deleted. (MTI index = 0x%X)</td>
</tr>
<tr>
<td>The MTunnel is not found</td>
<td>The MTI to be deleted was not found.</td>
</tr>
<tr>
<td>the MsgRPC_CMD, send to No.1 board, is 0x%X</td>
<td>Value of the RPC message sent to IO card 1.</td>
</tr>
<tr>
<td>the MsgRPC_CMD, Broadcast to IO board, is 0x%X</td>
<td>Value of the RPC message broadcast to all IO cards.</td>
</tr>
<tr>
<td>No.1 board, the Recive-MsgRPC_CMD is 0x%X</td>
<td>Value of the RPC message received on IO card 1.</td>
</tr>
<tr>
<td>No.1 board, the IO-Drv-CMD from mainboard is 0x%X</td>
<td>Value of driver context that IO card 1 received from the MPU.</td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the `debugging mtunnel ipc` command.

### Table 80 Output from the debugging mtunnel ipc command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recive-Msg is NULL</td>
<td>The received IPC message is null.</td>
</tr>
<tr>
<td>The length of receive-Msg is wrong</td>
<td>The size of the received IPC message is incorrect.</td>
</tr>
<tr>
<td>the MsgRPC_CMD, Send to No.1 Failed(Revlen error), is 0x%X</td>
<td>RPC communication with board 1 failed (incorrect length received). The command word is 0x%X.</td>
</tr>
<tr>
<td>the MsgRPC_CMD, Send to No.1 timeout, is 0x%X</td>
<td>RPC communication with board 1 failed (command timed out). The command word is 0x%X.</td>
</tr>
<tr>
<td>Alloc IPC_RPC_MSG error</td>
<td>RPC communication failed (memory allocation error).</td>
</tr>
</tbody>
</table>

Table 7 describes output fields and messages for the `debugging mtunnel management` command.
**Table 81 Output from the debugging mtunnel management command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can’t get MTunnel src addr</td>
<td>The MTI source address cannot be obtained.</td>
</tr>
<tr>
<td>MTunnel src addr is invalid</td>
<td>The obtained MTI source address is invalid (0).</td>
</tr>
<tr>
<td>Can’t get MTunnel dest address</td>
<td>The group IP address of the MTI cannot be obtained.</td>
</tr>
<tr>
<td>MTunnel dest addr is invalid</td>
<td>The obtained group IP address of the MTI is invalid (0).</td>
</tr>
<tr>
<td>The MTunnel has been</td>
<td>MTI state: Up or down.</td>
</tr>
<tr>
<td>MTunnel Src addr</td>
<td>MTI source address.</td>
</tr>
<tr>
<td>Set MTunnel Src addr</td>
<td>A MTI source address was set.</td>
</tr>
<tr>
<td>Set MTunnel group addr</td>
<td>A MTI group address was set.</td>
</tr>
</tbody>
</table>

Table 8 describes output fields and messages for the **debugging mtunnel packet** command.

**Table 82 Output from the debugging mtunnel packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The packet length is too short</td>
<td>Insufficient packet length.</td>
</tr>
<tr>
<td>Memory-continous is failed</td>
<td>The multicast tunnel failed to defragment data in the MBUF.</td>
</tr>
<tr>
<td>plp is NULL</td>
<td>The packet IP header address is null.</td>
</tr>
<tr>
<td>pstMGRE-Head is NULL</td>
<td>The packet GRE header address is null.</td>
</tr>
<tr>
<td>resv0 field error</td>
<td>The reserved 0 field of the GRE header is not 0.</td>
</tr>
<tr>
<td>check-sum error</td>
<td>Packet checksum result error.</td>
</tr>
<tr>
<td>version field error</td>
<td>The version field of the packet GRE header is not 0.</td>
</tr>
<tr>
<td>protocol field known</td>
<td>The packet payload is non-IPv4.</td>
</tr>
<tr>
<td>Decapsulate successfully</td>
<td>Packet was successfully de-encapsulated.</td>
</tr>
<tr>
<td>Payload is IP data</td>
<td>The VPN packet payload is IP data.</td>
</tr>
<tr>
<td>Payload is not IP data</td>
<td>The VPN packet payload is not IP data.</td>
</tr>
<tr>
<td>ADD GRE-head is failed</td>
<td>The multicast tunnel failed to add GRE header to packet.</td>
</tr>
<tr>
<td>ADD IP-head is failed</td>
<td>The multicast tunnel failed to add IP header to packet.</td>
</tr>
<tr>
<td>Can’t get mtunnel source address</td>
<td>The MTI source address cannot be obtained.</td>
</tr>
<tr>
<td>Src addr</td>
<td>MTI source address for packet was successfully encapsulated.</td>
</tr>
<tr>
<td>Can’t get mtunnel destination address</td>
<td>The MTI group IP address cannot be obtained.</td>
</tr>
<tr>
<td>Grp addr</td>
<td>The default group IP address of the MTI is being used.</td>
</tr>
<tr>
<td>Switch-Grp addr</td>
<td>The switch-group address of the MTI is being used.</td>
</tr>
<tr>
<td>MTunnel is not up</td>
<td>The MT to forward packets is not up.</td>
</tr>
<tr>
<td>The MTunnelMode-LLCoutput is NULL</td>
<td>The transmission function corresponding to the MT mode to forward the multicast packet is null.</td>
</tr>
</tbody>
</table>
A multicast tunnel interface (MTI) is a global interface. Unless otherwise stated, all examples in this section are based on centralized devices.

# Enable multicast tunnel error debugging. Output similar to the following example is generated when an IPv4 packet of another multicast tunnel is sent under the following conditions:

- A multicast tunnel is created between two PE devices.
- The tunnel parameters are configured at both ends to bring the tunnel interface up.

```bash
<Sysname> debugging mtunnel error
*Jan 23 16:30:17:06 2007 Sysname MTUNNEL/7/debug:
mtunnel_error: Can't get MTunnel PhyInfo Ptr.
```

// The multicast tunnel failed to obtain the physical statistics data block of the multicast tunnel interface.

# Enable multicast tunnel event debugging. Output similar to the following example is generated when a multicast tunnel is removed under the following conditions:

- The multicast tunnel is created between two PE devices
- Interrupts fail to be disabled when the multicast tunnel is removed.

```bash
<Sysname> debugging mtunnel event
*Jan 23 16:30:17:06 2007 Sysname MTUNNEL/7/debug:
mtunnel_event: SplImp is failed..
```

// The multicast tunnel failed to disable interrupts.

# Enable multicast tunnel management debugging. Output similar to the following example is generated under the following conditions:

- A multicast tunnel interface is created.
- The multicast tunnel group address is set to 235.0.0.1.

```bash
<Sysname> debugging mtunnel management
*Jan 23 15:52:15:02 2007 Sysname MTUNNEL/7/debug:
mtunnel_management: Set MTunnel group addr=235.0.0.1.
```

// The multicast tunnel set the multicast tunnel interface group address to 235.0.0.1.

```bash
*Jan 23 15:52:15:02 2007 Sysname MTUNNEL/7/debug:
mtunnel_management: Set MTunnel Src addr=11.11.11.11.
```

// The multicast tunnel set the multicast tunnel interface source address to 11.11.11.11.

```bash
*Jan 23 15:52:55:18 2007 DUT1 MTUNNEL/7/debug:
mtunnel_management: The MTunnel has been UP, no change.
```

// The link state of the multicast tunnel interface was up without any change.

# Enable multicast tunnel packet debugging. Output similar to the following example is generated under the following conditions:

- A multicast tunnel interface is created, with the default group address 235.0.0.1 and the source address 11.11.11.11.
- The MD is configured.
- The MD protocol is enabled to start protocol packet exchange.

```bash
<Sysname> debugging mtunnel packet
*Jan 23 15:52:15:02 2007 Sysname MTUNNEL/7/debug:
mgre_encapsulate: Payload is IP data.
```

// The packet that entered the multicast tunnel is an IP packet.
The multicast tunnel source address 11.11.11.11 is used for packet encapsulation.

The default group IP address 235.0.0.1 of the multicast tunnel is used for packet encapsulation.

The packet was successfully encapsulated.

The packet was successfully decapsulated.

Enable multicast tunnel IPC message debugging. When a multicast tunnel is created between two distributed PE devices, output similar to the following example is generated:

```
<Sysname> debugging mtunnel ipc
```

// The RPC message received from interface board 1 has an incorrect length.
MVRP debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging mvrp error

Use `debugging mvrp error` to enable MVRP error debugging.

Use `undo debugging mvrp error` to disable MVRP debugging.

Syntax

```
debugging mvrp error
undo debugging mvrp error
```

Default

MVRP error debugging is disabled.

Views

Use view

Default command level

1: Monitor level

Examples

```
# Enable MVRP error debugging.
<Sysname> debugging mvrp error
   // MVRP failed to send IPC messages to slot 1.
```

debugging mvrp event

Use `debugging mvrp event` to enable MVRP event debugging.

Use `undo debugging mvrp event` to disable MVRP event debugging.

Syntax

```
debugging mvrp event
undo debugging mvrp event
```

Default

MVRP event debugging is disabled.

Views

User view

Default command level

1: Monitor level
Examples

# Enable MVRP event debugging.
<Sysname> debugging mvrp event
*Mar 31 17:49:26:219 2011 Sysname MVRP/7/MVRP_DEBUG: EVENT:
VLAN list successfully created when the OP timed out.
*Mar 31 17:49:26:219 2011 Sysname MVRP/7/MVRP_DEBUG: EVENT:
VLAN list added to the trunk port.

// Creating VLAN 10 on the remote interface triggered the process of adding the VLAN list on the local interface to the trunk port.

debugging mvrp packet

Use `debugging mvrp packet` to enable MVRP packet debugging on an interface.
Use `undo debugging mvrp packet` to disable MVRP packet debugging on an interface.

Syntax

`debugging mvrp packet { send | receive } interface interface-type interface-number`
`undo debugging mvrp packet { send | receive } interface interface-type interface-number`

Default

MVRP packet debugging is disabled on an interface.

Views

User view

Default command level

1: Monitor level

Parameters

`send`: Specifies debugging for sent MVRP packets.
`receive`: Specifies debugging for received MVRP packets.
`interface interface-type interface-number`: Specifies an interface by its type and number.

Usage guidelines

Table 1 describes output fields and messages for the `debugging mvrp packet` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN Attribute</td>
<td>VLAN state:</td>
</tr>
<tr>
<td></td>
<td>• New—Declared and maybe not registered.</td>
</tr>
<tr>
<td></td>
<td>• JoinIn—Declared and registered.</td>
</tr>
<tr>
<td></td>
<td>• In—Not declared but registered.</td>
</tr>
<tr>
<td></td>
<td>• JoinEmpty—Declared but not registered.</td>
</tr>
<tr>
<td></td>
<td>• Empty—Not declared or registered.</td>
</tr>
<tr>
<td>VLAN ID</td>
<td>VLAN ID in the range of the minimum VLAN ID to the maximum VLAN ID that the system supports.</td>
</tr>
</tbody>
</table>
Examples

# Enable debugging for sent MVRP packets on Ethernet 1/1.
<Sysname> debugging mvrp packet send interface ethernet 1/1
*Mar 31 17:23:59:860 2011 Sysname MVRP/7/MVRP_DEBUG: PACKET.Ethernet1/1.TX:
  // MVRP sent packets through Ethernet 1/1.
  VLAN Attribute = JoinIn, VLAN ID = 1.
  // The VLAN ID was 1 and the attribute value was JoinIn.

debugging mvrp state

Use `debugging mvrp state interface` to enable MVRP state debugging for a VLAN on an interface.
Use `undo debugging mvrp state` to restore the default.

Syntax

debugging mvrp state interface interface-type interface-number vlan vlan-id
undo debugging mvrp state interface interface-type interface-number vlan vlan-id

Default

MVRP state debugging is disabled on an interface.

Views

User view

Default command level

1: Monitor level

Parameters

`interface-type interface-number`: Specifies an interface by its type and number.

Usage guidelines

Table 2 describes output fields and messages for the `debugging mvrp state` command.

Table 84 Output from the debugging mvrp state command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtrId</td>
<td>VLAN attribute ID, in the range of the minimum VLAN ID to the maximum VLAN ID that the system supports.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| APP   | Application state machine, which indicates the state of the attribute that the local participant declares to the remote participant. The state can be VO, VP, VN, AN, AA, QA, LA, AO, QO, AP, QP, or LO. Each state consists of two letters. The first letter indicates the state:  
  - V—Very anxious.  
  - A—Anxious.  
  - Q—Quiet.  
  - L—Leaving.  
  
  The second letter indicates the membership state:  
  - A—Active member.  
  - P—Passive member.  
  - O—Observer.  
  - N—New.  
  
  For example, VP indicates "Very anxious, Passive member." |
| Reg   | Registration state machine of the attribute declared by remote participants on the local participant. The state can be:  
  - IN—Registered.  
  - LV—Previously registered, but now being unregistered.  
  - MT—Not registered. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Event that triggers state machine transitions.</td>
</tr>
</tbody>
</table>

The application state machines include:
- **Begin!** — The state machine is initialized.
- **New!** — MRP requests the local participant to declare this attribute, and the tcDetected timer of the instance on the local participant is not 0.
- **Join!** — MRP requests the local participant to declare this attribute.
- **Lv!** — MRP requests the local participant to deregister this attribute.
- **rNew!** — The local participant receives a New message.
- **rJoinln!** — The local participant receives a Joinln message that the peer participant has registered.
- **rIn!** — The local participant receives an In message that the peer participant has registered, but not declared.
- **rJoinMt!** — The local participant receives a JoinMt message that the peer participant has not registered.
- **rMt!** — The local participant receives an Mt message that the peer participant has not registered or declared.
- **rLv!** — The local participant receives a Leave message.
- **rLA!** — The local participant receives a LeaveAll message.
- **Re-declare!** — When the MSTP port changes from a Desi port to a Root or Alte port, the process of re-declaring all MVRP attributes is triggered.
- **periodic!** — The Periodic timer expires.
- **tx!** — A packet ending timing is generated, and no LeaveAll event exists.
- **txLA!** — A packet sending timing is generated, and the LeaveAll flag is set.
- **txLA!** — A packet sending timing is generated, the LeaveAll flag is set, and the PDU message is full.

The registration state machines include:
- **Begin!** — The state machine is initialized.
- **rNew!** — The local participant receives a New message.
- **rJoinln!** — The local participant receives a Joinln message that the peer participant has registered.
- **rJoinMt!** — The local participant receives a JoinMt message that the peer participant has not registered.
- **rLv!** — The local participant receives a Leave message.
- **rLA!** — The local participant receives a LeaveAll message.
- **Re-declare!** — When the MSTP port changes from a Desi port to a Root or Alte port, the process of re-declaring all MVRP attributes is triggered.
Field | Description
---|---
LeaveAll | LeaveAll flag. This field is displayed only when the flag is true.

**Examples**

```plaintext
# Enable MVRP state debugging for VLAN 2 on Ethernet 1/1.
<Sysname> debugging mvrp state interface ethernet 1/1 vlan 2
*Mar 31 17:52:58:875 2011 Sysname MVRP/7/MVRP_DEBUG: FSM:
  Ethernet1/1: AttrID = 2: APP = VO Reg = IN, Event = rJoinMt!.
  // The application state machine was VO, the registration state machine was IN, and the event was rJoinMt!.
*Mar 31 17:52:58:938 2011 Sysname MVRP/7/MVRP_DEBUG: FSM:
  Ethernet1/1: AttrID = 2: APP = LO Reg = LV, Event = rLA!, LeaveAll = TRUE.
  // The application state machine was LO, the registration state machine was LV, the event was rLA!, and the LeaveAll flag was TRUE.
```
NAT debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

**debugging nat**

Use `debugging nat` to enable NAT debugging.

Use `undo debugging nat` to disable NAT debugging.

**Syntax**

```
debugging nat { alg | event | packet } [ interface interface-type interface-number ]
undo debugging nat { alg | event | packet } [ interface interface-type interface-number ]
```

**Default**

NAT debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- `alg`: Specifies ALG debugging. Support for this keyword depends on the device model.
- `event`: Specifies event debugging.
- `packet`: Specifies packet debugging.
- `interface interface-type interface-number`: Specifies NAT packet debugging for a specified interface.

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging nat alg` command.

**Table 85 Output from the debugging nat alg command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ftp Packet To Svr</td>
<td>NAT ALG processes FTP packets sent to the server.</td>
</tr>
<tr>
<td>(interface-type interface-number-out :)</td>
<td>The outbound packet is serviced by NAT on interface interface-type interface-number.</td>
</tr>
<tr>
<td>Find a Normal Ftp CMD</td>
<td>ALG received a normal FTP command.</td>
</tr>
<tr>
<td>Ftp Packet To Client</td>
<td>NAT ALG processes FTP packets sent to the client.</td>
</tr>
<tr>
<td>(interface-type interface-number-in :)</td>
<td>The inbound packet is serviced by NAT on interface interface-type interface-number.</td>
</tr>
</tbody>
</table>
Field Description
---
DNS A answer is translated successfully by dns-map(address1 -> address2). The IP address in the type A DNS reply has been translated.
- **address1** — The IP address before translation.
- **address2** — The IP address after translation.

Table 2 describes output fields and messages for the `debugging nat event` command.

### Table 86 Output from the debugging nat event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The packet can't translate by static.</td>
<td>Static address translation failed.</td>
</tr>
<tr>
<td>(interface-index-out)</td>
<td>The outbound packet is serviced by NAT on interface interface-index.</td>
</tr>
<tr>
<td>(interface-index-in)</td>
<td>The inbound packet is serviced by NAT on interface interface-index.</td>
</tr>
<tr>
<td>Ftp Packet To Svr</td>
<td>Processes FTP packets sent to the server using NAT ALG.</td>
</tr>
<tr>
<td>Translate Ftp PORT CMD fail</td>
<td>FTP port address translation failed.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the `debugging nat packet` command.

### Table 87 Output from the debugging nat packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>from VPN : vpn index</td>
<td>VPN index of the outbound packet (not displayed if no VPN exists).</td>
</tr>
<tr>
<td>is from NAT static</td>
<td>The outbound packet is translated by using static NAT.</td>
</tr>
<tr>
<td>( inside-source-ip: port1 - global-destination-ip: port2) ——&gt;</td>
<td>Before address translation:</td>
</tr>
<tr>
<td></td>
<td>• <strong>inside-source-ip</strong> — Source IP address.</td>
</tr>
<tr>
<td></td>
<td>• <strong>port1</strong> — Source port number.</td>
</tr>
<tr>
<td></td>
<td>• <strong>global-destination-ip</strong> — Public destination IP address.</td>
</tr>
<tr>
<td></td>
<td>• <strong>port2</strong> — Destination port number.</td>
</tr>
<tr>
<td>( global-source-ip: port1 - global-destination-ip: port2)</td>
<td>After address translation:</td>
</tr>
<tr>
<td></td>
<td>• <strong>global-source-ip</strong> — Source IP address.</td>
</tr>
<tr>
<td></td>
<td>• <strong>port1</strong> — Source port number.</td>
</tr>
<tr>
<td></td>
<td>• <strong>global-destination-ip</strong> — Public destination IP address.</td>
</tr>
<tr>
<td></td>
<td>• <strong>port2</strong> — Destination port number.</td>
</tr>
<tr>
<td>to VPN : vpn index</td>
<td>VPN index of the inbound packet (not displayed if no VPN exists).</td>
</tr>
<tr>
<td>is to NAT server</td>
<td>The inbound packet is translated by using the NAT Server feature.</td>
</tr>
<tr>
<td>is to NAT static</td>
<td>The inbound packet is translated by using static NAT.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(global-source-ip: port1 -</td>
<td>Before address translation:</td>
</tr>
<tr>
<td>global-destination-ip: port2)</td>
<td>• global-source-ip—Source IP address.</td>
</tr>
<tr>
<td></td>
<td>• port1—Source port number.</td>
</tr>
<tr>
<td></td>
<td>• global-destination-ip—Public</td>
</tr>
<tr>
<td></td>
<td>destination IP address.</td>
</tr>
<tr>
<td></td>
<td>• port2—Destination port number.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>(global-source-ip: port1 -</td>
<td>After address translation:</td>
</tr>
<tr>
<td>inside-destination-ip: port2)</td>
<td>• global-source-ip—Source IP address.</td>
</tr>
<tr>
<td></td>
<td>• port1—Source port number.</td>
</tr>
<tr>
<td></td>
<td>• inside-destination-ip—Destination</td>
</tr>
<tr>
<td></td>
<td>IP address.</td>
</tr>
<tr>
<td></td>
<td>• port2—Destination port number.</td>
</tr>
</tbody>
</table>

### Examples

# Enable NAT ALG debugging on the NAT device. When internal host 192.168.0.210 uses FTP to access public host 2.2.2.2, output similar to the following example is generated:

```
<Sysname> debugging nat alg
*Apr 26 12:05:36:881 2010 Sysname NAT/7/debug:Ftp Packet To Svr (Ethernet1/1-out :) Find a Normal Ftp CMD
// NAT used ALG to translate the address in the outgoing FTP packet sent to the server on Ethernet 1/1.
*Apr 26 12:05:36:952 2010 Sysname NAT/7/debug: Ftp Packet To Client (Ethernet1/1-in :) Find a Normal Ftp CMD
// NAT used ALG to translate the address in the incoming FTP packet sent to the client on Ethernet 1/1.
```

# Enable NAT event debugging on the NAT device. Output similar to the following example is generated when internal host 192.168.0.210 sends a packet to public host 2.2.2.2 under the condition that no matching static NAT entry exists:

```
<Sysname> debugging nat event
*Apr 26 12:05:37:33 2010 Sysname NAT/7/debug: "NAT_EVENT:(0x00001234-out)The packet can't translate by static.
// NAT failed to perform static address translation for the packet on interface 0x00001234.
```

# Enable NAT packet debugging on the NAT device. Output similar to the following example is generated when internal host 192.168.0.210 in VPN 10 pings public host 2.2.2.2 under the condition that a matching static NAT entry is configured:

```
<Sysname> debugging nat packet
*Apr 26 12:04:37:134 2010 Sysname NAT/7/debug:
(Ethernet1/1-out :) from VPN : 10 Pro : ICMP is from NAT static
( 192.168.0.210: --- - 2.2.2.2: ---) ---->
( 2.2.2.10:--- - 2.2.2.2: ---)
// NAT translated the source IP address of the outgoing packet on Ethernet 1/1 by using the static NAT entry.
```

```
*Apr 26 12:04:37:134 2010 Sysname NAT/7/debug:
(Ethernet1/1-in :)Pro : ICMP is to NAT static
( 2.2.2.2: --- - 2.2.2.10:---) ---->
( 2.2.2.2: --- - 192.168.0.210: ---)
// NAT translated the destination IP address of the incoming packet on Ethernet 1/1 by using the static NAT entry.
```
debugging userlog nat

Use debugging userlog nat to enable NAT logging debugging.
Use undo debugging userlog nat to disabled NAT logging debugging.

Syntax

detabling userlog nat
undo debugging userlog nat

Default

NAT logging debugging is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

NOTE:
Support for this command depends on the device model.

Table 4 describes output fields and messages for the debugging userlog nat command.

Table 88 Output from the debugging userlog nat command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Export-host address or syslog not configured. | The NAT logging function does not function correctly because either of the following reasons:  
• No log server is specified on the device.  
• The device is not configured to send logs to the information center.  
The log information is output in UDP packets. |
| Forced to output NAT logs. | The system is forced to output a certain number of NAT logs when the number of NAT logs exceeds 250000. |

Examples

# Enable NAT logging debugging on the NAT device. When NAT logging is enabled but no log server is specified on the device, output similar to the following example is generated:

<Sysname> debugging userlog nat
*Dec 6 15:36:38:956 2010 Sysname S8505 USERLOG/8/NAT:Slot=3;  
Export-host address or syslog not configured.  
// No log server is configured or the device is not configured to send logs to the information center.

# Enable NAT logging debugging on the NAT device. Output similar to the following example is generated under the conditions that NAT logging is enabled and a log server is specified:

<Sysname> debugging userlog nat
*Dec 6 15:36:39:03 2010 Sysname S8505 USERLOG/8/NAT:Slot=3;  
Exporting a UDP packet with 2 logs from slot 3.
// NAT logs were output in UDP packets from slot 3 to the log server. One UDP packet carried two NAT logs.

*Dec 6 15:36:39:54 2010 Sysname S8505 USERLOG/8/NAT:Slot=3; *
Forced to output NAT logs.

// The system was forced to output NAT logs because the NAT log buffer was full.
debugging natpt

Use **debugging natpt** to enable NAT-PT debugging.

Use **undo debugging natpt** to disable NAT-PT debugging.

**Syntax**

```
debugging natpt { alg | all | event | packet } [ interface interface-type interface-number ]
undo debugging natpt { alg | all | event | packet } [ interface interface-type interface-number ]
```

**Default**

NAT-PT debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

```
alg: Specifies the application layer gateway (ALG) debugging.
all: Specifies all types of debugging.
event: Specifies the event debugging.
packet: Specifies the packet debugging.
interface interface-type interface-number: Specifies an interface by its type and number.
```

**Examples**

**Figure 1 Network diagram**

```
natpt address-group 1 5.1.1.1 5.1.1.20
natpt prefix 1001::
natpt v6bound dynamic prefix 1001:: address-group 1 no-pat
interface Ethernet1/2
ip address 3.3.3.2 255.255.255.0
natpt enable
interface Ethernet1/1
ipv6 address 1000::2 64
natpt enable
```

- On single-core devices:
# Enable NAT-PT packet debugging on the NAT-PT device. When the IPv6 host uses FTP to access the IPv4 host, output similar to the following example is generated:

```
<sysname> debugging natpt packet
*0.1150141 sysname NATPT/7/debug:
  pro : TCP
  ttl : 127
  v6 pkt translated to v4 pkt
  (1000::0001 : 1041 - 1001::0303:0301 : 21)
  (5.1.1.1 : 1041 - 3.3.3.1 : 21)
// NAT-PT translated the source and destination IPv6 addresses in the received IPv6 TCP packet into IPv4 addresses.
*0.1150141 sysname NATPT/7/debug:
  pro : TCP
  ttl : 254
  v4 pkt translated to v6 pkt
  (3.3.3.1 : 21 - 5.1.1.1 : 1041)
  (1001::0303:0301 : 21 - 1000::0001 : 1041)
// NAT-PT translated the source and destination IPv4 addresses in the received IPv4 TCP packet into IPv6 addresses.
```

# Enable NAT-PT event debugging on the NAT-PT device. When the device receives an IPv4 packet, but no matching NAT-PT entry exists for the destination IPv4 address, output similar to the following example is generated:

```
<sysname> debugging natpt event
*0.478281 sysname NATPT/7/debug:
  mapping does not exist for the v4 destination address 111.255.255.255
// NAT-PT found no NAT-PT entry for IP address 111.255.255.255.
*0.478281 sysname NATPT/7/debug:
  the v4 pkt is not intended for NATPT. Returning the pkt back to V4 stack
// NAT-PT sent the IPv4 packet back to the IPv4 protocol stack because the packet was not intended for NAT-PT.
```

# Enable NAT-PT event debugging on the NAT-PT device. When the device receives an IPv6 packet, but no matching IPv6 prefix exists for the destination IPv6 address, output similar to the following example is generated:

```
<sysname> debugging natpt event
*0.825484 sysname NATPT/7/debug:
  v6 dest addr does not contain configured prefix
// NAT-PT did not find a matching prefix for the destination address of the IPv6 packet.
*0.825484 sysname NATPT/7/debug:
  the v6 pkt is not intended for NATPT. Returning the pkt back to V6 stack
// NAT-PT sent the IPv6 packet back to the IPv6 protocol stack because it was not intended for NAT-PT.
```

- On multi-core devices:
  # Enable NAT-PT packet debugging on the NAT-PT device. When the IPv6 host uses FTP to access the IPv4 host, output similar to the following example is generated:

```
<sysname> debugging natpt packet
*Mar 10 14:28:16:62 2008 sysname DPNATPT/7/PACKET:
  pro : TCP
```
TTL : 127
IPv6 packet is translated to IPv4 packet.
(1000::0001 : 1041 - 1001::0303:0301 : 21)
(5.1.1.1 : 1041 - 3.3.3.1 : 21)
// NAT-PT translated the source and destination IPv6 addresses in the received IPv6 TCP packet into IPv4 addresses.

*Mar 10 14:28:16:62 2008 Sysname DPNATPT/7/PACKET:
  Pro : TCP
  TTL : 254
IPv4 packet is translated to IPv6 packet.
(3.3.3.1 : 21 - 5.1.1.1 : 1041)
(1001::0303:0301 : 21 - 1000::0001 : 1041)
// NAT-PT translated the source and destination IPv4 addresses in the received IPv4 TCP packet into IPv6 addresses.

# Enable NAT-PT event debugging. Output similar to the following example is generated when the IPv6 host uses FTP to access the IPv4 host under the condition that NAT-PT is disabled on Ethernet 1/1:
<Sysname> debugging natpt event
*Mar 10 14:34:02:547 2008 Sysname DPNATPT/7/EVENT: The outgoing interface is disabled, dropping packet.
// NAT-PT discarded the packets because NAT-PT was disabled on the outgoing interface.

# Enable NAT-PT event debugging on the NAT-PT device. Output similar to the following example is generated when the IPv6 host uses FTP to access the IPv4 host under the condition that dynamic IPv6 NAT-PT mappings are deleted:
<Sysname> debugging natpt event
// NAT-PT discarded the packet because no IPv6 NAT-PT mappings were available.

# Enable NAT-PT ALG debugging on the NAT-PT device. Output similar to the following example is generated when the IPv6 host uses PORT to access the IPv4 host under the following conditions:
  o The IPv6 host has successfully accessed the IPv4 host by using FTP.
  o The sessions on the NAT-PT device have been deleted by using the reset session command.
  o The IPv6 NAT-PT dynamic mappings have been deleted.
<Sysname> debugging natpt alg
*Mar 10 14:45:10:562 2008 Sysname DPNATPT/7/ALG: Failed to process command EPRT.
// Because FTP control sessions and IPv6 dynamic NAT-PT mappings were removed from Device, Device could not translate addresses in the PORT commands sent from the IPv6 host.
The output description tables in this document only contain fields and messages that require an explanation.

### debugging ipv6 nd detection packet

Use `debugging ipv6 nd detection packet` to enable ND packet debugging for ND detection.

Use `undo debugging ipv6 nd detection packet` to disable ND packet debugging for ND detection.

**Syntax**

```
debugging ipv6 nd detection packet
undo debugging ipv6 nd detection packet
```

**Default**

ND packet debugging of ND detection is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Usage guidelines**

Table 1 describes the output fields and messages for the `debugging ipv6 nd detection packet` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Received: Port</td>
<td>Port that received an ND packet.</td>
</tr>
<tr>
<td>VLAN</td>
<td>VLAN that received the packet.</td>
</tr>
<tr>
<td>DstMACInEth</td>
<td>Destination MAC address in the Ethernet frame header of the packet.</td>
</tr>
<tr>
<td>SrcMACInEth</td>
<td>Source MAC address in the Ethernet frame header of the packet.</td>
</tr>
<tr>
<td>PktType</td>
<td>Type of the packet.</td>
</tr>
<tr>
<td>SrcIP</td>
<td>Source IPv6 address of the packet.</td>
</tr>
<tr>
<td>DstIP</td>
<td>Destination IPv6 address of the packet.</td>
</tr>
<tr>
<td>TrgIP</td>
<td>Target address carried in an NS packet, an NA packet, or a redirect packet.</td>
</tr>
<tr>
<td>RRDstIP</td>
<td>Destination address carried in a redirect packet.</td>
</tr>
<tr>
<td>LLA</td>
<td>MAC address in the source link-layer address option field of the ND packet.</td>
</tr>
</tbody>
</table>
### Field Description

| Dropped ND Packet for no entry match | The ND packet was dropped because the ND detection function failed to find any address binding entry matching the packet during source verification. |

### Examples

# Enable ND packet debugging for ND detection.

```
<Sysname> debugging ipv6 nd detection packet
*Jun 23 16:36:16:719 2009 Sysname ND/7/ND DETECTION_PACKET:
Received: Port= Ethernet 1/1, VLAN= 1, DstMACInEth= 0015-e943-820e, SrcMACInEth= 3333-ff00-0001, PktType= 133, SrcIP= 5::4, DstIP= 5::1, TrgIP= ::, RRDstIP= ::, LLA= 0000-0000-0002
// An ND packet was received.
*Jun 23 16:36:16:734 2009 Sysname ND/7/ND DETECTION_PACKET:
Dropped ND Packet for no entry match.
// The ND packet was dropped because no matching address entry was found.
```
NetStream debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging ip netstream event

Use `debugging ip netstream event` to enable NetStream event debugging.

Use `undo debugging ip netstream event` to disable NetStream event debugging.

Syntax

```
debugging ip netstream event
undo debugging ip netstream event
```

Default

NetStream event debugging is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

Table 1 describes the output fields and messages for the `debugging ip netstream event` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Stream Create cache</td>
<td>NetStream created IP stream cache.</td>
</tr>
<tr>
<td>IP Stream Destroy cache</td>
<td>NetStream cleared IP stream cache.</td>
</tr>
<tr>
<td>IP Stream Reset cache by force</td>
<td>NetStream reset IP stream cache.</td>
</tr>
<tr>
<td>IP Stream Malloc from system</td>
<td>NetStream allocated memory for IP stream cache.</td>
</tr>
<tr>
<td>IP Stream Add to active list</td>
<td>NetStream added an IP stream to the active list.</td>
</tr>
<tr>
<td>IP Stream Add to age list</td>
<td>NetStream added an IP stream to the aged list.</td>
</tr>
<tr>
<td>IP Stream Add a stream (Now 1 active streams)</td>
<td>NetStream added an IP stream entry (a total of one active entry).</td>
</tr>
<tr>
<td>IP Stream Update a stream</td>
<td>NetStream updated an IP stream entry.</td>
</tr>
<tr>
<td>IP Stream Age a stream (tcp fin or rst)</td>
<td>An IP stream entry (TCP FIN or RST) aged out.</td>
</tr>
<tr>
<td>IP Stream Age a stream (active age or overflow)</td>
<td>An IP stream entry aged out due to active flow aging or entry overflow.</td>
</tr>
<tr>
<td>IP Stream Age 1 streams (inactive timeout)</td>
<td>An IP stream entry aged out due to inactive flow aging.</td>
</tr>
<tr>
<td>IP Stream Age 1 streams (age all)</td>
<td>An IP stream entry aged out due to forced aging.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>IP Stream Age 1 streams (cache full)</td>
<td>An IP stream entry aged out due to entry overflow.</td>
</tr>
<tr>
<td>IP Stream Age all streams (cache modify)</td>
<td>All IP stream entries aged out due to cache modification.</td>
</tr>
<tr>
<td>IP Stream Age all streams (reset)</td>
<td>All IP stream entries aged out due to resetting.</td>
</tr>
<tr>
<td>IP Stream Export a stream</td>
<td>NetStream exported an IP stream entry.</td>
</tr>
<tr>
<td>IP Stream Export 0 packets for timeout (no host or no template)</td>
<td>No IP packets were exported because of timeout. This event typically occurs because no destination host is configured or no template is activated.</td>
</tr>
<tr>
<td>IP Stream Export 0 packets for cache full (no host or no template)</td>
<td>No IP packets were exported because the cache was full. This event typically occurs because no destination host is configured or no template is activated.</td>
</tr>
<tr>
<td>IP Stream Export 1 packets for timeout</td>
<td>An IP packet was exported due to timeout.</td>
</tr>
<tr>
<td>IP Stream Export 1 packets for cache full</td>
<td>An IP packet was exported because the cache was full.</td>
</tr>
<tr>
<td>AS Aggre Create cache</td>
<td>NetStream created AS aggregation stream cache.</td>
</tr>
<tr>
<td>AS Aggre Destory cache when 0 entrys left</td>
<td>NetStream cleared the AS aggregation stream cache when it did not contain entries.</td>
</tr>
<tr>
<td>AS Aggre Add a stream (Now 1 active streams)</td>
<td>NetStream added an AS aggregation stream entry (a total of one active entry).</td>
</tr>
<tr>
<td>AS Aggre Update a stream</td>
<td>NetStream updated an AS aggregation stream entry.</td>
</tr>
<tr>
<td>AS Aggre Export a stream</td>
<td>NetStream exported an AS aggregation stream entry.</td>
</tr>
<tr>
<td>AS Aggre Export 0 packets for timeout (no host or no template)</td>
<td>No AS aggregation packets were exported because of timeout. This event typically occurs because no destination host is configured or no template is activated.</td>
</tr>
<tr>
<td>AS Aggre Export 0 packets for aggre cache full (no host or no template)</td>
<td>No AS aggregation packets were exported because the cache was full. This event typically occurs because no destination host is configured or no template is activated.</td>
</tr>
<tr>
<td>AS Aggre Export 1 packets for timeout</td>
<td>An AS aggregation packet was exported due to timeout.</td>
</tr>
<tr>
<td>AS Aggre Export 1 packets for aggre cache full</td>
<td>An AS aggregation packet was exported because the cache was full.</td>
</tr>
<tr>
<td>Add to send queue (Now queue length is 1) Packet Type: Normal IP Version No: 9 Records: 1</td>
<td>A packet was added to packet export queue 1. The packet type is single IP stream, the packet format is v9, and the number of IP stream entries is 1.</td>
</tr>
<tr>
<td>Send 1 packets (enter queue)</td>
<td>A packet was sent and entered a packet export queue.</td>
</tr>
</tbody>
</table>
| Succeed in sending (Now queue length is 0) Packet Type: Normal IP Version No: 9 Records: 1 | A packet was successfully sent:  
  - Packet export queue is 0.  
  - Packet type is single IP stream  
  - Packet format is v9.  
  - Number of IP stream entries is 1. |
Fail to send (Now queue length is 0)
Packet Type: Normal   IP Version No: 9  Records: 1
- NetStream failed to send a packet.
  - Packet export queue is 0.
  - Packet type is single IP stream.
  - Packet format is v9.
  - Number of IP stream entries is 1.

Active Template (Now 1 active templates): protocol-port inbound Id 271
A template was activated (a total of one activated template). The template type is protocol-port inbound and template ID is 271.

Deactive Template (Now 1 active templates): protocol-port inbound Id 271
A template was deactivated (a total of one activated template). The template type is protocol-port inbound and template ID is 271.

Export Template : protocol-port inbound Id 271
A template was exported. The template type is protocol-port inbound and template ID is 271.

Select Template for stream (Version 9 Type IP outbound Id 282)
A v9 template was selected for a single stream entry. The template type is IP outbound and template ID is 282.

Select Template for aggre (Version 9 Type destination-prefix inbound Id 261)
A v9 template was selected for an aggregation stream entry. The template type is destination-prefix inbound and template ID is 261.

CPU 0   IP Stream        Add to active list:
Direct: 0 Stream Type: IP IP version: 4
InIf: 0 OutIf: 1048577 InVrf: 0 OutVrf: 0
SrcIP: 0.0.0.0 DstIP: 0.0.0.0 Prot: 1
SrcPort: 0 DstPort: 2048 Tol: 0x0 TcpFlag: 0x0
SrcAS: 0 DstAS: 0 SrcMask: 0 DstMask: 0
Nexthop: 0.0.0.0 BGP Nexthop: 0.0.0.0
Label 1:0-0-0 2:0-0-0 3:0-0-0
TopLabel Type: UNKNOWN, IP: 0.0.0.0, Mask: 0.0.0.0
SrcMAC: 0000-0000-0000 DstMAC: 0000-0000-0000 SrcVlan: 0 DstVlan: 0
First: 0 Last: 0 Pkts: 0 0 Bytes: 0 84
Sample Mode: 0 Sample Interval: 0

IP stream entry fields:
- Direct—Direction.
- InIf—Inbound interface number.
- OutIf—Outbound interface number.
- InVrf—VPN that inbound packets belong.
- OutVrf—VPN that outbound packets belong.
- SrcIP—Source IP address.
- DstIP—Destination IP address.
- Prot—Protocol.
- SrcPort—Source port number.
- DstPort—Destination port number.
- SrcAS—Source AS number.
- DstAS—Destination AS number.
- SrcMask—Source mask.
- DstMask—Destination mask.
- TopLabel Type—Type of the label at the top of the label stack.
- SrcMAC—Source MAC address.
- DstMAC—Destination MAC address.
- SrcVlan—Source VLAN ID.
- DstVlan—Destination VLAN ID.
- First—First active time of a stream.
- Last—Last active time of a stream.
- Pkts—Number of packets in a stream.
- Bytes—Number of bytes of a stream.
### Field Description

**CPU 0 AS Aggre**  Add a stream (Now 1 active streams):

- **InIf:** 0  **OutIf:** 1048576  **SrcAS:** 0  **DstAS:** 0
- **Direct:** 0  **Aggre Type:** AS IP version : 4
- **First:** 551894110  **Last:** 551894110  **Streams:** 1  **Pkts:** 0  **Bytes:** 84
- **Sample Mode:** FULL  **Sample Interval:** 0

#### AS aggregation fields:

- **InIf**—Inbound interface number.
- **OutIf**—Outbound interface number.
- **SrcAS**—Source AS number.
- **DstAS**—Destination AS number.
- **Aggre Type**—Aggregation type.
- **First**—First active time of a stream.
- **Last**—Last active time of a stream.
- **Streams**—Number of streams for aggregation statistics.
- **Pkts**—Number of packets in a stream.
- **Bytes**—Number of bytes of a stream.

### Examples

The output in the following examples is generated when the following conditions exist:
- NetStream statistics are configured in the inbound direction of the interface Ethernet 1/0 on a device.
- A destination host is configured.
- AS aggregation is enabled.
- Version 9 is configured.
- The inactive timer is set to 10 seconds.

# Enable IPv4 NetStream event debugging. Output in the following example is generated when an IP packet arrives at Ethernet 1/0:

```bash
<Sysname> debugging ip netstream event
*Mar 21 12:31:47:343 2008 H3C NS/7/NS_EVENT:
CPU 0   IP Stream        Add to active list:
Direct: I Stream Type:   IP IP version: 4
InIf: 51314696 OutIf: 0 InVrf: 0 OutVrf: 0
SrcPort: 0 DstPort: 2048 Tos: 0x0 TcpFlag: 0x0
SrcAS: 0 DstAS: 0 SrcMask: 24 DstMask: 32
TopLabel Type: UNKNOWN IP: 0.0.0.0 Mask: 0.0.0.0
Nexthop: 0.0.0.0 BGP Nexthop: 0.0.0.0
Lable 1:0-0-0 2:0-0-0 3:0-0-0
First: 0 Last: 0 Pkts: 0 0 Bytes: 0 60
Sample Mode: 0 Sample Interval: 0
```

// An IPv4 stream was added to the list.

```bash
*Mar 21 12:31:47:343 2008 H3C NS/7/NS_EVENT:
CPU 0   IP Stream        Add a stream (Now 1 active streams):
Direct: I Stream Type:   IP IP version: 4
InIf: 51314696 OutIf: 118816768 InVrf: 0 OutVrf: 0
SrcPort: 0 DstPort: 2048 Tos: 0x0 TcpFlag: 0x0
SrcAS: 0 DstAS: 0 SrcMask: 24 DstMask: 32
```

126
Nexthop: 127.0.0.1 BGP Nexthop: 0.0.0.0
Lable 1:0-0-0 2:0-0-0 3:0-0-0
TopLabel Type: UNKNOWN IP: 0.0.0.0 Mask: 0.0.0.0
SrcMAC: 0000-0000-0000 DstMAC: 0000-0000-0000 SrcVlan: 0 DstVlan: 0
First: 10981810 Last: 10981810 Pkts: 0 1 Bytes: 0 60
Sample Mode: 0 Sample Interval: 0

// An IPv4 stream was added to the list.

# Enable IPv4 NetStream event debugging. Output in the following example is generated when NetStream entry aged out after 10 seconds because the NetStream inactive aging timer timed out.
<Sysname> debugging ip netstream event

*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
CPU 0 IP Stream Age 1 streams (inactive timeout)

// An IP stream entry aged out due to inactive flow aging.
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
Select Template for stream(Version 9 Type IP inbound Id 281 )

// A template was selected for the IP stream entry.
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
CPU 0 IP Stream Export a stream:
Direct: I Stream Type: IP IP version: 4
InIf: 51314696 OutIf: 118816768 InVrf: 0 OutVrf: 0
SrcPort: 0 DstPort: 2048 Tos: 0x0 TcpFlag: 0x0
SrcAS: 0 DstAS: 0 SrcMask: 24 DstMask: 32
Nexthop: 127.0.0.1 BGP Nexthop: 0.0.0.0
Lable 1:0-0-0 2:0-0-0 3:0-0-0
TopLabel Type: UNKNOWN IP: 0.0.0.0 Mask: 0.0.0.0
SrcMAC: 0000-0000-0000 DstMAC: 0000-0000-0000 SrcVlan: 0 DstVlan: 0
First: 10981810 Last: 10981810 Pkts: 0 1 Bytes: 0 60
Sample Mode: 0 Sample Interval: 0

// An IP stream entry was exported.
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
CPU 0 as Aggre Add a stream (Now 1 active streams):
InIf: 51314696 OutIf: 118816768 SrcAS: 0 DstAS: 0
Direct: I Aggre Type: as IP version: 4
First: 10981810 Last: 10981810 Streams: 1 Pkts: 0 1 Bytes: 0 60
Sample Mode: FULL Sample Interval: 0

// An AS aggregation stream entry was added (a total of one active entry).
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
Add to send queue (Now queue length is 1):
Packet Type: Normal IP Version No: 9 Records: 1

// An IP packet was added to the packet queue.
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
Send 1 packets (enter queue)

// NetStream was sending a packet.
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
Succeed in sending (Now queue length is 0):
Packet Type: Normal   IP  Version No: 9  Records: 1

// A packet was successfully sent.
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
CPU 0   IP Stream        Export 1 packets for timeout

// An IP stream entry packet was sent.
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
Select Template for aggre(Version 9 Type as inbound Id 257)

// A template was selected for an AS aggregation stream entry.
*Mar 21 12:31:56:890 2008 H3C NS/7/NS_EVENT:
CPU 0 as Aggre   Export a stream:
  InIf: 51314696 OutIf: 118816768 SrcAS: 0 DstAS: 0
  Direct: I Aggre Type: as IP version : 4
  First: 10981810 Last: 10981810 Streams: 1 Pkts: 0 1 Bytes: 0 60
  Sample Mode: FULL Sample Interval: 0
*Mar 21 12:31:56:906 2008 H3C NS/7/NS_EVENT:
CPU 0 as Aggre   Export 1 streams

// An AS aggregation stream entry was exported.
*Mar 21 12:31:56:906 2008 H3C NS/7/NS_EVENT:
Add to send queue (Now queue length is 1):
Packet Type: Aggre as  Version No: 9  Records: 1

// An AS aggregation packet was added to the packet export queue.
*Mar 21 12:31:56:906 2008 H3C NS/7/NS_EVENT:
Send 1 packets (enter queue)

// NetStream was sending a packet.
*Mar 21 12:31:56:906 2008 H3C NS/7/NS_EVENT:
Succeed in sending (Now queue length is 0):
Packet Type: Aggre as  Version No: 9  Records: 1

// A packet was successfully sent.
*Mar 21 12:31:56:906 2008 H3C NS/7/NS_EVENT:
CPU 0 as Aggre   Export 1 packets for timeout

// An AS aggregation packet was sent.

debugging ip netstream packet

Use debugging ip netstream packet to enable NetStream packet debugging.
Use undo debugging ip netstream packet to disable NetStream packet debugging.

Syntax

ddebugging ip netstream packet
undo debugging ip netstream packet

Default

NetStream packet debugging is disabled.

Views

User view
**Default command level**

1: Monitor level

**Usage guidelines**

Table 2 describes the output fields and messages for the `debugging ip netstream packet` command.

**Table 91 Output from the debugging ip netstream packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send succeed!</td>
<td>NetStream sent a packet successfully.</td>
</tr>
<tr>
<td>Send fail!</td>
<td>NetStream failed to send a packet.</td>
</tr>
</tbody>
</table>

Packet Type: Normal   IP  Version No: 5  Records: 2

- **Packet Type**—Exported packet type.
- **Version No**—Exported packet version.
- **Records**—Number of packets exported.
- **SrcIP(Port)**—Source IP address. The field in parentheses shows the source port number.
- **DstIP(Port)**—Destination IP address. The field in parentheses shows the destination port number.
- **VrfID**—Private network route index.

**Examples**

# Enable IPv4 NetStream packet debugging. Output similar to the following example is generated when an IP packet arrives at Ethernet 1/0 and an entry ages out under the following conditions:

- NetStream is enabled on Ethernet 1/0.
- A destination host is configured.

```
<Sysname> debugging ip netstream packet
*Mar 21 09:41:07:953 2008 H3C NS/7/NS_PACKET:
Send succeed! Packet Type: Normal   IP  Version No: 5  Records: 2
// A version 5 IP stream was sent successfully.
```
NQA debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging nqa

Use `debugging nqa` to enable NQA debugging.
Use `undo debugging nqa` to disable NQA debugging.

Syntax

```
debugging nqa { all | error | event | reaction }
undo debugging nqa { all | error | event | reaction }
```

Default

NQA debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- **all**: Enables all types of debugging for NQA.
- **error**: Enables NQA error debugging.
- **event**: Enables NQA event debugging.
- **reaction**: Enables debugging for NQA collaboration entries.

Usage guidelines

Table 1 describes output fields and messages for the `debugging nqa event` command.

Table 2 describes output fields and messages for the `debugging nqa reaction` command.
### Table 93 Output from the debugging nqa reaction command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner: owner</td>
<td>Name of the administrator who creates the NQA operation.</td>
</tr>
<tr>
<td>Tag: tag</td>
<td>NQA operation tag.</td>
</tr>
<tr>
<td>Previous status: status</td>
<td>- 1—The previous status is invalid.</td>
</tr>
<tr>
<td></td>
<td>- 2—The number of consecutive probe failures exceeds the threshold.</td>
</tr>
<tr>
<td></td>
<td>- 3—The operation succeeded.</td>
</tr>
<tr>
<td>Current status: status</td>
<td>- 1—The current status is invalid.</td>
</tr>
<tr>
<td></td>
<td>- 2—The number of consecutive probe failures exceeds the threshold.</td>
</tr>
<tr>
<td></td>
<td>- 3—The operation succeeded.</td>
</tr>
</tbody>
</table>

### Examples

```bash
# Enable all types of debugging for NQA.
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging nqa all

# Create an NQA operation.
<Sysname> system-view
[Sysname] nqa entry admin test
*Apr 29 21:44:18:32 2007 Sysname NQA/7/NQA_Event: NQA entry(admin-test) is created.
   // An NQA operation was created. The administrator name is admin and operation tag is test.

# Configure the operation type as ICMP echo, and specify the destination address for the ICMP echo operation.
[Sysname-nqa-admin-test] type icmp-echo
[Sysname-nqa-admin-test-icmp-echo] destination ip 10.2.2.1
[Sysname-nqa-admin-test-icmp-echo] quit

# Schedule an NQA operation.
[Sysname] nqa schedule admin test start-time now lifetime forever
*Apr 29 21:47:25:630 2007 Sysname NQA/7/NQA_Event: NQA schedule: Entry(admin-test) is added to the test schedule queue.
   // NQA added the NQA operation to the scheduling queue.
   // The NQA operation was successfully scheduled.
   // NQA started the ICMP echo operation.
   // NQA started to send packets.
```

// The operation succeeded.

# Cancel the NQA operation schedule.

[Sysname] undo nqa schedule admin test

*Apr 29 21:49:00:206 2007 Sysname NQA/7/NQA_Event: NQA Schedule: The schedule of entry(admin-test) is cancelled.

// The NQA operation schedule was successfully cancelled.
NTP debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

**debugging ntp-service**

Use `debugging ntp-service` to enable NTP debugging.

Use `undo debugging ntp-service` to disable NTP debugging.

**Syntax**

```
debugging ntp-service { access | adjustment | all | authentication | event | filter | packet | parameter | refclock | selection | synchronization | validity }
undo debugging ntp-service { access | adjustment | all | authentication | event | filter | packet | parameter | refclock | selection | synchronization | validity }
```

**Default**

All NTP debugging functions are disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **access**: NTP access control debugging.
- **adjustment**: NTP clock adjustment debugging.
- **all**: All types of debugging for NTP.
- **authentication**: NTP authentication debugging.
- **event**: NTP event debugging.
- **filter**: NTP filter debugging.
- **packet**: NTP packet debugging.
- **parameter**: Debugging for NTP clock parameters.
- **refclock**: Debugging for NTP reference clock.
- **selection**: Debugging for NTP clock selection information.
- **synchronization**: Debugging for NTP clock synchronization.
- **validity**: Debugging for NTP remote host validity.

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging ntp-service adjustment` command.
### Table 94 Output from the debugging ntp-service adjustment command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTP: gradual systime</td>
<td>NTP is gradually adjusting the system time.</td>
</tr>
<tr>
<td>NTP: step systime</td>
<td>NTP is adjusting the system time in one step.</td>
</tr>
<tr>
<td>adj: string</td>
<td>Time that is adjusted in the current time adjustment operation.</td>
</tr>
<tr>
<td>residual: string</td>
<td>Remaining value for the most recent time adjustment.</td>
</tr>
<tr>
<td>offset: string</td>
<td>Offset for step adjustment.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the **debugging ntp-service authentication** command.

### Table 95 Output from the debugging ntp-service authentication command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session_key</td>
<td>Session key.</td>
</tr>
<tr>
<td>srcadr: string</td>
<td>Source IP address.</td>
</tr>
<tr>
<td>dstadr: string</td>
<td>Destination IP address.</td>
</tr>
<tr>
<td>keyid: string</td>
<td>Key ID.</td>
</tr>
<tr>
<td>life: string</td>
<td>Lifetime of a key.</td>
</tr>
<tr>
<td>auth_agekeys</td>
<td>Parameters related to key lifetime.</td>
</tr>
<tr>
<td>time: string</td>
<td>Time period from when the system starts until the current time.</td>
</tr>
<tr>
<td>trusted keynum: string</td>
<td>Number of trusted keys.</td>
</tr>
<tr>
<td>expired keynum: string</td>
<td>Number of expired keys.</td>
</tr>
<tr>
<td>Authentication keyID: string</td>
<td>Authentication key ID.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the **debugging ntp-service event** command.

### Table 96 Output from the debugging ntp-service event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTP: control event</td>
<td>NTP control event.</td>
</tr>
<tr>
<td>event: string</td>
<td>Event code.</td>
</tr>
<tr>
<td>eventnum: string</td>
<td>Number of events.</td>
</tr>
<tr>
<td>peer: string</td>
<td>IP address of the peer.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the **debugging ntp-service filter** command.

### Table 97 Output from the debugging ntp-service filter command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTP: adj freq</td>
<td>Adjustment frequency.</td>
</tr>
<tr>
<td>last clockoffset: string</td>
<td>Last clock offset.</td>
</tr>
<tr>
<td>last drift_comp: string</td>
<td>Last frequency.</td>
</tr>
<tr>
<td>new clockOffset: string</td>
<td>New clock offset.</td>
</tr>
</tbody>
</table>
new drift_comp: string

New frequency.

Table 5 describes output fields and messages for the **debugging ntp-service packet** command.

**Table 9.8 Output from the debugging ntp-service packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| NTP: titleAndTip control packet from sourceIPAddress to DestIPAddress | • titleAndTip—Title and prompt information.  
  • sourceIPAddress—Source IP address of the packet.  
  • DestIPAddress—Destination IP address of the packet. |
| version: string            | Version number in the control packet. |
| r: string                  | Response bit in the control packet. |
| e: string                  | Error bit in the control packet. |
| m: string                  | More bit in the control packet. |
| o: string                  | Operation code in the control packet. |
| sequence: string           | Sequence number in the control packet. |
| status: string             | Status words in the control packet. |
| associationID: string      | Association ID in the control packet. |
| data: string               | Data information in the control packet. |
| authenticator: string      | Authenticator information in the control packet. |
| packet to string           | Destination IP address of the packet. |
| leap: string               | Leap indicator in a packet. |
| version: string            | Version number in a packet. |
| mode: string               | Working mode in a packet. |
| vrfindex: string           | Index of the VPN that received or sent a packet. |
| stratum: string            | Stratum information in a packet. |
| poll: string               | Poll interval information in a packet. |
| precision: string          | Precision information in a packet. |
| rdel: string               | Root delay information in a packet. |
| rdsp: string               | Root dispersion information in a packet. |
| refid: string              | ID of the reference clock.  
  If the reference clock is a local clock, the contents of this field vary  
  by the stratum of the local clock. If the stratum is 1, this field  
  displays **LOCL**. If the stratum is another value, this field displays  
  the IP address of the local clock.  
  If the reference clock is the clock of another device in the network,  
  this field displays the IP address of this device. |
<p>| reftime: string            | Reference timestamp. |
| orgtime: string            | Originate timestamp. |
| rectime: string            | Receive timestamp. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xmttime: string</td>
<td>Transmit timestamp.</td>
</tr>
<tr>
<td>inptime: string</td>
<td>Timestamp for processing a packet.</td>
</tr>
</tbody>
</table>

Packet from SourceIPAddress to DestIPAddress on InterfaceName:
- **SourceIPAddress**—Source IP address.
- **DestIPAddress**—Destination IP address.
- **InterfaceName**—Name of the interface receiving the packet.

Table 6 describes output fields and messages for the debugging ntp-service parameter command.

### Table 99 Output from the debugging ntp-service parameter command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTP: popcorn spike: string</td>
<td>Offset jitter.</td>
</tr>
<tr>
<td>NTP: discard: string</td>
<td>The sample is discarded if the lifetime of the new sample is lower than or equal to the selected sample. This field displays the lifetime of the new sample.</td>
</tr>
<tr>
<td>clock_filter(PeerAddr, SampleOffset, SampleDelay, SampleDisp)</td>
<td>IP address, sample offset, sample delay, and sample dispersion of the peer in clock-filter.</td>
</tr>
<tr>
<td>offset: string</td>
<td>Offset of the peer.</td>
</tr>
<tr>
<td>delay: string</td>
<td>Delay of the peer.</td>
</tr>
<tr>
<td>dispersion: string</td>
<td>Dispersion of the peer.</td>
</tr>
<tr>
<td>std: string</td>
<td>Jitter of the peer.</td>
</tr>
</tbody>
</table>

Table 7 describes output fields and messages for the debugging ntp-service refclock command.

### Table 100 Output from the debugging ntp-service refclock command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Event:</td>
<td>Reference clock event.</td>
</tr>
<tr>
<td>Clock: string</td>
<td>IP address of the reference clock.</td>
</tr>
<tr>
<td>Event: string</td>
<td>Description of the clock event.</td>
</tr>
<tr>
<td>Code: string</td>
<td>Event code of the clock.</td>
</tr>
</tbody>
</table>

RefClock Transmit: At CurrentTime IPAddr
- **CurrentTime**—Current system time.
- **IPAddr**—IP address of the reference clock.

RefClock Sample:
- Sample of the reference clock.

sampleNum: string
- Number of samples.

offset: string
- Offset.

disp: string
- Dispersion.

std: string
- Jitter.

RefClock Receive: At CurrentTime IPAddr
- **CurrentTime**—Current system time.
- **IPAddr**—IP address of the reference clock.
Table 8 describes output fields and messages for the debugging ntp-service selection command.

**Table 101 Output from the debugging ntp-service selection command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nlist: string</td>
<td>Number of the candidate clocks in candidate clock list.</td>
</tr>
<tr>
<td>allow: string</td>
<td>Number of surviving candidates.</td>
</tr>
<tr>
<td>found: string</td>
<td>Number of discarded candidates.</td>
</tr>
<tr>
<td>low: string</td>
<td>Lower value of the sample offset.</td>
</tr>
<tr>
<td>high: string</td>
<td>Upper value of the sample offset.</td>
</tr>
<tr>
<td>candidate: string</td>
<td>IP address of a candidate clock.</td>
</tr>
<tr>
<td>cdist: string</td>
<td>Root distance of a candidate clock.</td>
</tr>
<tr>
<td>disp: string</td>
<td>Dispersion of a candidate clock.</td>
</tr>
<tr>
<td>survivor: string</td>
<td>IP address of the candidate clock survived in a check.</td>
</tr>
<tr>
<td>offset: string</td>
<td>Offset of the candidate clock survived in a check.</td>
</tr>
<tr>
<td>cdist: string</td>
<td>Dispersion of the candidate clock survived in a check.</td>
</tr>
<tr>
<td>syspeep: string</td>
<td>IP address of the clock selected by the system.</td>
</tr>
<tr>
<td>offset: string</td>
<td>Offset of the clock selected by the system.</td>
</tr>
<tr>
<td>Root-distance</td>
<td>Synchronization distance.</td>
</tr>
<tr>
<td>Max root-distance</td>
<td>Maximum synchronization distance selected by the clock source.</td>
</tr>
<tr>
<td>SYS POLL</td>
<td>System polling interval.</td>
</tr>
</tbody>
</table>

Table 9 describes output fields and messages for the debugging ntp-service validity command.

**Table 102 Output from the debugging ntp-service validity command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTP: packet from SourceIPAddr, TestResult validity tests TestCode</td>
<td></td>
</tr>
<tr>
<td>SourceIPAddr</td>
<td>Source IP address of the packet.</td>
</tr>
<tr>
<td>TestResult</td>
<td>Test result, successful or failed.</td>
</tr>
<tr>
<td>TestCode</td>
<td>Code of the object to be tested.</td>
</tr>
<tr>
<td>Code and its meaning of the object to be tested:</td>
<td></td>
</tr>
<tr>
<td>0x0001</td>
<td>Duplicate information.</td>
</tr>
<tr>
<td>0x0002</td>
<td>False information.</td>
</tr>
<tr>
<td>0x0004</td>
<td>Unsynchronized information.</td>
</tr>
<tr>
<td>0x0008</td>
<td>Peer delay/skew dispersion.</td>
</tr>
<tr>
<td>0x0010</td>
<td>Peer authentication failure.</td>
</tr>
<tr>
<td>0x0020</td>
<td>Unsynchronized peer clock.</td>
</tr>
<tr>
<td>0x0040</td>
<td>Peer stratum level exceeds the highest value.</td>
</tr>
<tr>
<td>0x0080</td>
<td>Root delay/dispersion exceeds the highest value.</td>
</tr>
<tr>
<td>0x0100</td>
<td>No authentication on the peer.</td>
</tr>
<tr>
<td>0x0200</td>
<td>Access denied.</td>
</tr>
</tbody>
</table>

137
Examples

# Enable NTP packet debugging on Device A. The output in this example was created when the following conditions exist:

- On Device A and Device B, the IP addresses for VLAN-interface 1 are 10.1.1.1 and 10.1.1.2, respectively.
- The IP addresses can be pinged from each other.
- Device B’s local clock is to be used as a reference source, with the stratum level of 8.
- Device A is configured to synchronize its clock to Device B in client/server mode.

<DeviceA> debugging ntp-service packet
<DeviceA> terminal debugging
<DeviceA> terminal monitor

// Log monitoring on the current terminal was enabled.
*Aug 12 06:04:42:129 2005 DeviceA NTP/7/debug_NTP_packet_xmt:
   packet to 10.1.1.2
   leap: 3, version: 3, mode: 3, vrfindex: 0
   stratum: 0, poll: 64, precision: 2^18
   rdel: 0.000, rdsp: 0.000, refid: 0.0.0.0
   orgtime: 00:00:00.000 UTC Jan 1 1900(00000000.00000000)
   rectime: 00:00:00.000 UTC Jan 1 1900(00000000.00000000)
   xmttime: 06:04:42.127 UTC Aug 12 2005(C6A6BA7A.20B039EF)

// NTP sent an NTP time request to Device B at 10.1.1.2.
%Oct 11 14:42:45:139 2006 DeviceA NTP/4/NTP_LOG:
System leap changes from 3 to 0 after clock update.
%Oct 11 14:42:45:140 2006 DeviceA NTP/4/NTP_LOG:
System stratum changes from 16 to 9 after clock update.

// Log information when the clock alarm and clock stratum of the system changed was generated.
*Oct 11 14:42:45:141 2006 DeviceA NTP/7/debug_NTP_packet_rcv:
   packet from 10.1.1.2 to 10.1.1.1 on Vlan-interface1
   leap: 0, version: 3, mode: 4, vrfindex: 0
   stratum: 8, poll: 64, precision: 2^18
   rdel: 0.000, rdsp: 10.941, refid: 127.127.1.0
   orgtime: 06:04:42.127 UTC Aug 12 2005(C6A6BA7A.20B039EF)
   rectime: 14:42:45.132 UTC Oct 11 2006(C8D78165.21F88FC9)
   xmttime: 14:42:45.132 UTC Oct 11 2006(C8D78165.22075F6F)
   inptime: 06:04:42.137 UTC Aug 12 2005(C6A6BA7A.23316E37)

// Device A at 10.1.1.1 received the NTP response from Device B at 10.1.1.2.

NOTE:
This example provides only the packet exchange process for the first two packets.
debugging oaa remote-interface

Use `debugging oaa remote-interface` to enable debugging for the OAA remote interface management module.

Use `undo debugging oaa remote-interface` to disable debugging for the OAA remote interface management module.

**Syntax**

```
debugging oaa remote-interface { all | error | event }
undo debugging oaa remote-interface { all | error | event }
```

**Default**

Debugging is disabled for the OAA remote interface management module.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- `all`: All types of debugging for the OAA remote interface management module.
- `error`: Error debugging for the OAA remote interface management module.
- `event`: Event debugging for the OAA remote interface management module.

**Examples**

```
# Enable debugging for the remote interface management module. The output in this example was created when the following conditions exist:
• OAA remote interface management is configured on the client.
• SNMP client is configured to communicate with the SNMP server.
• Remote interface management is enabled.
• The interface information is added to the dynamic remote interface table.
• The Refresh button on the remote interface management page is clicked.
<Sysname> debugging oaa remote-interface all
*Mar 20 14:50:50:313 2008 Sysname RMIF/7/DEBUG RMIF EVENT:
Update snapshot remote-interface table.
// The snapshot remote interface table was updated.
```
The output description tables in this document only contain fields and messages that require an explanation.

**OSPF debugging commands**

**debugging ospf event**

Use **debugging ospf event** to enable OSPF event debugging.

Use **undo debugging ospf event** to disable OSPF event debugging.

**Syntax**

```
debugging ospf [ process-id ] event [ bfd | error | graceful-restart | interface | neighbor ]
undo debugging ospf [ process-id ] event [ bfd | error | graceful-restart | interface | neighbor ]
```

**Default**

OSPF event debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **process-id**: Specifies an OSPF process by its ID, in the range of 1 to 65535.
- **bfd**: Specifies OSPF BFD event debugging.
- **error**: Specifies OSPF error event debugging.
- **graceful-restart**: Specifies OSPF graceful restart (GR) event debugging.
- **interface**: Specifies interface event debugging.
- **neighbor**: Specifies OSPF neighbor event debugging.

**Usage guidelines**

If no process ID is specified, the event debugging information of all the OSPF processes will be displayed.

Table 1 describes output fields and messages for the **debugging ospf event bfd** command.

**Table 103 Output from the debugging ospf event bfd command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF process-id:OSPFF received packet having conflicted Router ID :rt-id</td>
<td>OSPF received a packet with a conflicting Router ID. rt-id indicates the router ID of the neighbor.</td>
</tr>
</tbody>
</table>
Table 2 describes output fields and messages for the `debugging ospf event neighbor` command.

### Table 104: Output from the debugging ospf event neighbor command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF process-id</td>
<td>OSPF process ID.</td>
</tr>
<tr>
<td>Nbr nbr-ip</td>
<td>Detailed information of neighbor state changes:</td>
</tr>
<tr>
<td>Rcv nbr-event</td>
<td>• nbr-ip—Neighbor interface IP address.</td>
</tr>
<tr>
<td>State original-state -&gt; current-state</td>
<td>• nbr-event—Event triggering the HelloReceived, Start, 2WayReceived, NegotiationDone, ExchangeDone, BadLSReq, LoadingDone, AdjOK?, 1-Way, KillInbr, Inactivity Timer, or LLDown.</td>
</tr>
<tr>
<td></td>
<td>• original-state/current-state—Previous and current neighbor state. The state can be Down, Attempt, Init, 2Way, ExStart, Exchange, Loading, or Full.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable OSPF interface event debugging on Router A. The output in this example was created when the following conditions exist:
- Ethernet 1/0 (150.1.1.24) on Router A is connected to Ethernet 1/0 (150.1.1.24) on Router B over a broadcast network.
- On Router A, OSPF process 1 is created and area 0 is created in OSPF process 1. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
- On Router B, OSPF process 1 is created. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.

```
<RouterA> debugging ospf event interface
%Dec 12 09:24:58:978 2006 RouterA IFNET/4/UPDOWN:
Line protocol on the interface Ethernet1/0 is UP
OSPF 1: Intf 150.1.1.1 Rcv InterfaceUp State Down -> Waiting.
// Interface state was changed from Down to Waiting.
OSPF 1: Intf 150.1.1.1 Rcv BackupSeen State Waiting -> BackupDR.
// Interface state was changed from Waiting to BackupDR.
```

# Enable OSPF neighbor event debugging on Router A. The output in this example was created when the following conditions exist:
- Ethernet 1/0 (150.1.1.24) on Router A is connected to Ethernet 1/0 (150.1.1.24) on Router B over a broadcast network.
• On Router A, OSPF process 1 is created and area 0 is created in OSPF process 1. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.

• On Router B, OSPF process 1 is created. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.

<RouterA> debugging ospf event neighbor

OSPF 1: Nbr 150.1.1.2 Rcv KillNbr State Full -> Down.
OSPF 1: Nbr 150.1.1.2 Rcv HelloReceived State Down -> Init.
OSPF 1: Nbr 150.1.1.2 Rcv 2WayReceived State Init -> 2Way.
OSPF 1: Nbr 150.1.1.2 Rcv AdjOk State 2Way -> ExStart.
OSPF 1: Nbr 150.1.1.2 Rcv NegotiationDone State ExStart -> Exchange.
OSPF 1: Nbr 150.1.1.2 Rcv ExchangeDone State Exchange -> Loading.
OSPF 1: Nbr 150.1.1.2 Rcv LoadingDone State Loading -> Full.

// OSPF process 1 established an adjacency relationship with neighbor 150.1.1.2

# Enable OSPF GR event debugging on Router A. The output in this example was created when the following conditions exist:

• Ethernet 1/0 (150.1.1.1/24) on Router A is connected to Ethernet 1/0 (150.1.1.2/24) on Router B over a broadcast network.

• On Router A:
  - OSPF process 1 is created and area 0 is created in OSPF process 1.
  - Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
  - OSPF process 1 is enabled with OSPF local link signaling, OSPF out-of-band synchronization, and OSPF GR.

• On Router B:
  - OSPF process 1 is created.
  - Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
  - OSPF process 1 is enabled with OSPF local link signaling and OSPF out-of-band synchronization.

When you restart GR in OSPF process 1, output similar to the following example is generated:

<RouterA> reset ospf 1 process graceful-restart

%Dec 12 09:36:12:500 2006 RouterA RM/3/RMLOG:OSPF-NBRCHANGE: Process 1, Neighbour 150.1.1.2(Ethernet1/0) from Full to Down
OSPF 1: Intf 150.1.1.1 Rcv InterfaceDown State BackupDR -> Down.
OSPF 1 nonstandard GR Started for OSPF Router

// The device started OSPF GR.
OSPF 1 notify RM that OSPF process will enter GR.
OSPF 1 created GR wait timer, timeout interval is 40(s).
OSPF 1 created GR Interval timer, timeout interval is 120(s).
OSPF 1: Intf 150.1.1.1 Rcv InterfaceUp State Down -> Waiting.
OSPF 1: Intf 150.1.1.1 Rcv BackupSeen State Waiting -> BackupDR.
OSPF 1 created OOB Progress timer for neighbor 150.1.1.2.
OSPF 1 restarted OOB Progress timer for neighbor 150.1.1.2.
OSPF 1 restarted OOB Progress timer for neighbor 150.1.1.2.

%Dec 12 09:36:12:566 2006 RouterA RM/3/RMLOG:OSPF-NBRCHANGE: Process 1, Neighbour 150.1.1.2(Ethernet1/0) from Loading to Full
debugging ospf hot-standby

Use `debugging ospf hot-standby` to enable hot-standby debugging for OSPF.

Use `undo debugging ospf hot-standby` to disable hot-standby debugging for OSPF.

Syntax

```
debugging ospf [ process-id ] hot-standby
undo debugging ospf [ process-id ] hot-standby
```

Default

OSPF hot-standby debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `process-id`: Specifies an OSPF process by its ID, in the range of 1 to 65535.

Usage guidelines

If no process ID is specified, the hot-standby debugging information of all the OSPF processes will be displayed.

Table 3 describes output fields and messages for the `debugging ospf hot-standby` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF HSB Bkp: Instance ID:</td>
<td>OSPF backed up the OSPF process data under the specified instance.</td>
</tr>
<tr>
<td>instance-id</td>
<td>• <code>instance-id</code> — OSPF instance ID.</td>
</tr>
<tr>
<td>Process ID: process-id</td>
<td>• <code>process-id</code> — OSPF process ID.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| OSPF HSB Bkp: Instance ID: *instance-id* Process ID: *process-id* Area ID: *area-id* | OSPF backed up the OSPF process area data under the specified instance.  
  - *instance-id*—OSPF instance ID.  
  - *process-id*—OSPF process ID.  
  - *area-id*—Area ID. |
| OSPF HSB Res: Instance ID: *instance-id* Process ID: *process-id* | OSPF restored the OSPF process data under the specified instance.  
  - *instance-id*—OSPF instance ID.  
  - *process-id*—OSPF process ID. |

### Examples

# Enable hot-standby debugging for OSPF process 1. The output in this example was created when the following conditions exist:

- Router A is a dual-master device and its master board boots first.
- On Router A, OSPF process 1 is created, and area 0 and area 1 are configured under OSPF process 1.
- Area 0 is configured to contain network 10.10.10.10/32. OSPF is enabled on Loopback 0 (10.10.10.10/32).
- Area 1 is configured to contain network 11.11.11.11/32. OSPF is enabled on Loopback 1 (11.11.11.11/32).

When you insert the standby board, output similar to the following example is generated:

```
<RouterA> debugging ospf 1 hot-standby
%Apr 26 10:45:44:348 2007 RouterA HA/4/LOG: Batch backup started
OSPF 1 Back-up process data Start.
*Apr 26 10:45:44:363 2007 RouterA RM/6/RMDEBUG:
OSPF HSB Bkp: Instance ID: 0  Process ID: 1
OSPF 1 Back-up process data End.
// OSPF started and then ended backing up OSPF process 1 data.
OSPF 1 Back-up Area data Start.
*Apr 26 10:45:44:379 2007 RouterA RM/6/RMDEBUG:
OSPF HSB Bkp: Instance ID: 0  Process ID: 1  Area ID: 0.0.0.0
OSPF 1 Back-up Area data End.
// OSPF started and then ended backing up data for area 0 of OSPF process 1.
OSPF 1 Back-up Area data Start.
*Apr 26 10:45:44:394 2007 RouterA RM/6/RMDEBUG:
OSPF HSB Bkp: Instance ID: 0  Process ID: 1  Area ID: 0.0.0.1
OSPF 1 Back-up Area data End.
// OSPF started and then ended backing up data backup for area 1 of OSPF process 1.
*Apr 26 10:45:44:394 2007 RouterA RM/6/RMDEBUG:
```
debugging ospf lsa

Use `debugging ospf lsa` to enable OSPF LSA debugging.

Use `undo debugging ospf lsa` to disable OSPF LSA debugging.

Syntax

```
depugging ospf [ process-id ] lsa [ generate | install ]
undo debugging ospf [ process-id ] lsa [ generate | install ]
```

Default

OSPF LSA debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `process-id`: Specifies an OSPF process by its ID in the range of 1 to 65535.
- `generate`: Specifies LSA generation debugging.
- `install`: Specifies debugging for the installation of LSAs into the LSDB.

Usage guidelines

If no process ID is specified, the LSA debugging information of all the OSPF processes will be displayed.

Table 4 describes output fields and messages for the `debugging ospf lsa` command.

Table 106 Output from the `debugging ospf lsa` command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF process-id</td>
<td>OSPF process ID.</td>
</tr>
</tbody>
</table>
| op-type LSA at x ms| Operation on LSAs:
  - Generate—Generating LSAS
  - Install—Installing LSAs. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSAType: <code>ls-type</code> LinkStateId: <code>link-state-id</code> Advertising Rtr: <code>rt-id</code></td>
<td>LSA header information:</td>
</tr>
<tr>
<td></td>
<td>• <code>ls-type</code>—LSA type:</td>
</tr>
<tr>
<td></td>
<td>o 1 for Router LSA.</td>
</tr>
<tr>
<td></td>
<td>o 2 for network LSA.</td>
</tr>
<tr>
<td></td>
<td>o 3 for net-summary LSA.</td>
</tr>
<tr>
<td></td>
<td>o 4 for ASBR-summary LSA.</td>
</tr>
<tr>
<td></td>
<td>o 5 for AS-external–LSA.</td>
</tr>
<tr>
<td></td>
<td>o 7 for NSSA LSA.</td>
</tr>
<tr>
<td></td>
<td>o 9, 10, and 11 for Opaque LSA.</td>
</tr>
<tr>
<td></td>
<td>• <code>link-state-id</code>—LSA ID.</td>
</tr>
<tr>
<td></td>
<td>• <code>rt-id</code>—ID of the advertising router.</td>
</tr>
<tr>
<td>LSA Age: <code>age</code> Options : ExRouting: ON/OFF</td>
<td>LSA header information:</td>
</tr>
<tr>
<td></td>
<td>• <code>age</code>—LSA age.</td>
</tr>
<tr>
<td></td>
<td>• <code>ON/OFF</code>—Indicates whether external routing is supported.</td>
</tr>
<tr>
<td>Length: <code>ls-len</code> Seq: <code>seq-num</code> CheckSum: <code>checksum</code></td>
<td>LSA header information:</td>
</tr>
<tr>
<td></td>
<td>• <code>ls-len</code>—LS length.</td>
</tr>
<tr>
<td></td>
<td>• <code>seq-num</code>—LS sequence number.</td>
</tr>
<tr>
<td></td>
<td>• <code>checksum</code>—Checksum of the entire LSA except for the LSA age field.</td>
</tr>
<tr>
<td>Capabilities: VBit: <code>EBit</code> BBit: <code>NtBit</code> Link#: <code>link-count</code> TOS#: <code>tos-num</code> Metric <code>cost</code></td>
<td>Router LSA contents:</td>
</tr>
<tr>
<td></td>
<td>• <code>VBit</code>—0x40 for a virtual link.</td>
</tr>
<tr>
<td></td>
<td>• <code>EBit</code>—0x200 for an External LSA.</td>
</tr>
<tr>
<td></td>
<td>• <code>BBit</code>—0x100 for an ABR.</td>
</tr>
<tr>
<td></td>
<td>• <code>NtBit</code>—0x1000 for a router that unconditionally translates Type-7 LSAs to Type-5 LSAs.</td>
</tr>
<tr>
<td></td>
<td>• <code>Link-count</code>—Number of links in the Router LSA.</td>
</tr>
<tr>
<td></td>
<td>• <code>tos-num</code>—Number of TOSs in the Router LSA.</td>
</tr>
<tr>
<td></td>
<td>• <code>cost</code>—Link cost.</td>
</tr>
<tr>
<td>Net Mask: <code>net-mask</code> Attached Router: <code>rt-id</code></td>
<td>Network LSA contents:</td>
</tr>
<tr>
<td></td>
<td>• <code>net-mask</code>—Network mask.</td>
</tr>
<tr>
<td></td>
<td>• <code>rt-id</code>—ID of a neighbor.</td>
</tr>
<tr>
<td>Net Mask: <code>net-mask</code> Metric: <code>cost</code></td>
<td>Contents of Summary and ASBR-Summary LSAs:</td>
</tr>
<tr>
<td></td>
<td>• <code>net-mask</code>—Network mask.</td>
</tr>
<tr>
<td></td>
<td>• <code>cost</code>—Link cost.</td>
</tr>
<tr>
<td></td>
<td>• <code>net-mask</code>—Network mask.</td>
</tr>
<tr>
<td></td>
<td>• <code>tos</code>—Type of Service.</td>
</tr>
<tr>
<td></td>
<td>• <code>cost</code>—Link cost.</td>
</tr>
<tr>
<td></td>
<td>• <code>fwd-addr</code>—Forwarding address.</td>
</tr>
<tr>
<td></td>
<td>• <code>rt-tag</code>—External route tag.</td>
</tr>
</tbody>
</table>
Examples

# Enable debugging for the installation of LSAs into the LSDB on Router A. The output in this example was created when the following conditions exist:

- Ethernet 1/0 (150.1.1.1/24) on Router A is connected to Ethernet 1/0 (150.1.1.2/24) on Router B over a broadcast network.
- On Router A, OSPF process 1 is created and area 0 is created in OSPF process 1. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
- On Router B, OSPF process 1 is created. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.

```
<RouterA> debugging ospf lsa install
*Sep 8 17:51:02:234 2006 RouterA RM/6/RMDEBUG:OSPF 1: Install LSA at 4796222 ms:
*Sep 8 17:51:02:244 2006 RouterA RM/6/RMDEBUG:LSAType: 1.
*Sep 8 17:51:02:244 2006 RouterA RM/6/RMDEBUG:LinkStateId: 201.1.1.1.
*Sep 8 17:51:02:254 2006 RouterA RM/6/RMDEBUG:LSA Age: 0 Options: ExRouting:ON.
*Sep 8 17:51:02:254 2006 RouterA RM/6/RMDEBUG:Length: 36 Seq# 80000008 CheckSum: 60445.
*Sep 8 17:51:02:254 2006 RouterA RM/6/RMDEBUG:Capabilities: VBit:0 EBit: 512 BBit: 0 NtBit: 0 Link# 1.
*Sep 8 17:51:02:254 2006 RouterA RM/6/RMDEBUG:LinkID: 150.1.1.0 LinkData: 255.255.255.0 LinkType: 3.
*Sep 8 17:51:02:254 2006 RouterA RM/6/RMDEBUG:TOS# 0 Metric 10.
```

// OSPF process 1 installed the Router-LSAs it generated.
```
*Sep 8 17:51:06:766 2006 RouterA RM/6/RMDEBUG:OSPF 1: Install LSA at 4800748 ms:
*Sep 8 17:51:06:776 2006 RouterA RM/6/RMDEBUG:LSA Age: 5 Options: ExRouting:ON.
```

// OSPF process 1 installed the Router-LSAs generated by the peer.
```
*Sep 8 17:51:06:766 2006 RouterA RM/6/RMDEBUG:OSPF 1: Install LSA at 4800748 ms:
```

// Because the local router is a DR, OSPF process 1 installed the Network-LSAs it generated.
```
*Sep 8 17:51:07:238 2006 RouterA RM/6/RMDEBUG:OSPF 1: Install LSA at 4801229 ms:
```

147
// OSPF process 1 installed the Router-LSAs it generated, with stub link changed to transit link.

// OSPF process 1 installed the Router-LSAs generated by the peer into the LSDB, with stub link changed to transit link.

// OSPF process 1 installed type-5 LSAs, and included the redistributed route 123.1.1.0 255.255.255.0.

# Enable debugging for LSA generation on Router A. The output in this example was created when the following conditions exist:

- Ethernet 1/0 (150.1.1.1/24) on Router A is connected to Ethernet 1/0 (150.1.1.2/24) on Router B over a broadcast network.
- On Router A, OSPF process 1 is created and area 0 is created in OSPF process 1. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
- On Router B, OSPF process 1 is created. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.

<RouterA> debugging ospf lsa generate
debugging ospf mpls-te

Use `debugging ospf mpls-te` to enable debugging for OSPF MPLS traffic engineering.

Use `undo debugging ospf mpls-te` to disable debugging for OSPF MPLS traffic engineering.

**Syntax**

```
debugging ospf [ process-id ] mpls-te
undo debugging ospf [ process-id ] mpls-te
```

**Default**

Debugging for OSPF MPLS traffic engineering is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

`process-id`: Specifies an OSPF process by its ID, in the range of 1 to 65535.

**Usage guidelines**

If no process ID is specified, the MPLS traffic engineering debugging information of all the OSPF processes will be displayed.
Examples

# Enable MPLS-TE debugging for OSPF on Router A. The output in this example was created when the following conditions exist:

- Router A is configured with MPLS TE, OSPF, and LDP. Loopback 0 is created on Router A, with an IP address of 1.1.1.1/32. Ethernet 1/0 (150.1.1.1/24) on Router A is connected to Ethernet 1/0 (150.1.1.2/24) on Router B over a broadcast network. On Router A:
  - OSPF process 1 is created and is enabled with Opaque.
  - Area 0 is created in OSPF process 1 and is enabled with TE.
  - Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.

- Router B is also configured with MPLS TE, CSPF, and LDP. Loopback 0 is created on Router B, with an IP address of 2.2.2.2/32. On Router B:
  - OSPF process 1 is created and is enabled with Opaque.
  - Area 0 is created in OSPF process 1 and is enabled with TE.
  - Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.

- A TE tunnel interface is created on Router A, with an IP address of 6.1.1.1/24. The TE tunnel interface runs MPLS TE and the destination is 2.2.2.2.

```
<RouterA> debugging ospf 1 mpls-te
150.1.1.2(Ethernet1/1) from Loading to Full
*Dec 20 15:01:37:86 2006 RouterA RM/6/RMDEBUG:OSPF TE
  NSM_LoadingDone: LOADING DONE
*Dec 20 15:01:37:86 2006 RouterA RM/6/RMDEBUG:OSPF TE
  HandleNeighborStateChg: BROADCAST
*Dec 20 15:01:37:86 2006 RouterA RM/6/RMDEBUG:OSPF TE
  GetBroadCastNewGenNbr: Nbr DR, NbrIp = 2.2.2.2
*Dec 20 15:01:37:86 2006 RouterA RM/6/RMDEBUG:OSPF TE
  GetBroadCastNewGenNbr: SUCCESS, MultiCastTeNbrID = 2.2.2.2
*Dec 20 15:01:37:86 2006 RouterA RM/6/RMDEBUG:OSPF TE
  Received Link Info from RM
  Process : 1
  Area : 0.0.0.0
  Link Type : 4
  TE Metric : 0
  Max BW : 0
  Max Resv BW : 0
  Admin Grp : 0

Received Network LSA by OSPF TE
  Process ID : 1
  Area ID : 0.0.0.0
  Link Id : 150.1.1.2
  Attached Rtr Count : 2
  Advtg Router ID : 2.2.2.2

*Dec 20 15:01:37:86 2006 RouterA RM/6/RMDEBUG:OSPF TE
  Send Network LSA to CSPF SUCCESS
  Process ID : 1
```
Area ID : 0.0.0.0
Attached Rtr Count : 2
MPLS DR Router ID : 2.2.2.2
DR Intf Addr : 150.1.1.2

Received Network LSA by OSPF TE
Process ID : 1
Area ID : 0.0.0.0
Link Id : 150.1.1.2
Attached Rtr Count : 2
Advtg Router ID : 2.2.2.2

*Dec 20 15:01:37:98 2006 RouterA RM/6/RMDEBUG:OSPF TE
Send Network LSA to CSPF SUCCESS
Process ID : 1
Area ID : 0.0.0.0
Attached Rtr Count : 2
MPLS DR Router ID : 2.2.2.2
DR Intf Addr : 150.1.1.2

*Dec 20 15:01:37:118 2006 RouterA RM/6/RMDEBUG:OSPF TE
UpdateNetworkLsa - MODIFY
*Dec 20 15:01:37:118 2006 RouterA RM/6/RMDEBUG:OSPF TE
Received Network LSA by OSPF TE
Process ID : 1
Area ID : 0.0.0.0
Link Id : 150.1.1.2
Attached Rtr Count : 2
Advtg Router ID : 2.2.2.2

*Dec 20 15:01:37:118 2006 RouterA RM/6/RMDEBUG:OSPF TE
Send Network LSA to CSPF SUCCESS
Process ID : 1
Area ID : 0.0.0.0
Attached Rtr Count : 2
MPLS DR Router ID : 2.2.2.2
DR Intf Addr : 150.1.1.2

Received Network LSA by OSPF TE
Process ID : 1
Area ID : 0.0.0.0
Link Id : 150.1.1.2
Attached Rtr Count : 2
Advtg Router ID : 2.2.2.2

*Dec 20 15:01:37:130 2006 RouterA RM/6/RMDEBUG:OSPF TE
Send Network LSA to CSPF SUCCESS
Process ID : 1
Area ID             : 0.0.0.0
Attached Rtr Count  : 2
MPLS DR Router ID   : 2.2.2.2
DR Intf Addr        : 150.1.1.2

*Dec 20 15:01:37:130 2006 RouterA RM/6/RMDEBUG:OSPF TE
ReceiveOpqLsa: Processed Success

*Dec 20 15:01:40:194 2006 RouterA RM/6/RMDEBUG:OSPF TE
Received Network LSA by OSPF TE
Process ID          : 1
Area ID             : 0.0.0.0
Link Id             : 3.1.1.2
Attached Rtr Count  : 2
Advtg Router ID     : 3.3.3.3

*Dec 20 15:01:50:194 2006 RouterA RM/6/RMDEBUG:OSPF TE
Received Network LSA by OSPF TE
Process ID          : 1
Area ID             : 0.0.0.0
Link Id             : 3.1.1.2
Attached Rtr Count  : 2
Advtg Router ID     : 3.3.3.3

*Dec 20 15:01:52:194 2006 RouterA RM/6/RMDEBUG:OSPF TE
Received Network LSA by OSPF TE
Process ID          : 1
Area ID             : 0.0.0.0

**debugging ospf non-stop-routing**

* Use `debugging ospf non-stop-routing` to enable OSPF NSR debugging.
* Use `undo debugging ospf non-stop-routing` to disable OSPF NSR debugging.

**Syntax**

`debugging ospf non-stop-routing`

`undo debugging ospf non-stop-routing`

**Default**

OSPF NSR debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Usage guidelines**

Table 5 describes output fields and messages for the `debugging ospf non-stop-routing` command.
### Table 107 Output from the debugging ospf non-stop-routing command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth start</td>
<td>Active/standby switchover started.</td>
</tr>
<tr>
<td>Smooth start on the interface</td>
<td>Active/standby switchover started on the interface.</td>
</tr>
<tr>
<td>Back up in real time removal</td>
<td>OSPF NSR is backing up the specified neighbor to be removed in real time.</td>
</tr>
<tr>
<td>neighbor neighbor of interface index in area area-id of process process-id</td>
<td></td>
</tr>
<tr>
<td>• neighbor—Neighbor ID.</td>
<td></td>
</tr>
<tr>
<td>• index—Logical index of the interface.</td>
<td></td>
</tr>
<tr>
<td>• area-id—Area ID.</td>
<td></td>
</tr>
<tr>
<td>• process-id—Process ID.</td>
<td></td>
</tr>
<tr>
<td>Back up in batches removal</td>
<td>OSPF NSR is backing up the specified neighbor to be removed in batches.</td>
</tr>
<tr>
<td>neighbor neighbor of interface index in area area-id of process process-id</td>
<td></td>
</tr>
<tr>
<td>• neighbor—Neighbor ID.</td>
<td></td>
</tr>
<tr>
<td>• index—Logical index of the interface.</td>
<td></td>
</tr>
<tr>
<td>• area-id—Area ID.</td>
<td></td>
</tr>
<tr>
<td>• process-id—Process ID.</td>
<td></td>
</tr>
<tr>
<td>HA acknowledge flushed lsa type: type, advertising router: advrtr, link ID: linkid</td>
<td>HA acknowledged deleting the LSA.</td>
</tr>
<tr>
<td>• type—LSA type.</td>
<td></td>
</tr>
<tr>
<td>• advrtr—Advertising router.</td>
<td></td>
</tr>
<tr>
<td>• linkid—Link ID.</td>
<td></td>
</tr>
<tr>
<td>HA acknowledge the same lsa type: type, advertising router: advrtr, link ID: linkid, age: age, sequence number: seq</td>
<td>HA acknowledged the same LSA type.</td>
</tr>
<tr>
<td>• type—LSA type.</td>
<td></td>
</tr>
<tr>
<td>• advrtr—Advertising router.</td>
<td></td>
</tr>
<tr>
<td>• linkid—Link ID.</td>
<td></td>
</tr>
<tr>
<td>• age—LSA age.</td>
<td></td>
</tr>
<tr>
<td>• seq—Sequence number.</td>
<td></td>
</tr>
<tr>
<td>HA acknowledge update lsa type: type, advertising router: advrtr, link ID: linkid, age: age, sequence number: seq</td>
<td>HA acknowledged updating the LSA.</td>
</tr>
<tr>
<td>• type—LSA type.</td>
<td></td>
</tr>
<tr>
<td>• advrtr—Advertising router.</td>
<td></td>
</tr>
<tr>
<td>• linkid—Link ID.</td>
<td></td>
</tr>
<tr>
<td>• age—LSA age.</td>
<td></td>
</tr>
<tr>
<td>• seq—Sequence number.</td>
<td></td>
</tr>
</tbody>
</table>

### Examples

# Enable OSPF NSR debugging on Router A. The output in this example was created when the following conditions exist:

- On Router A, both the active MPU and standby MPU start. On Router A, OSPF process 1 is created and area 0 is created in OSPF process 1. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
  - OSPF process 1 is created.
  - Area 0 is created in OSPF process 1.
  - Area 0 contains network 3.1.1.0/32, and Area 1 contains network 4.1.1.0/32.
  - Two OSPF interfaces are configured.
- Router B is the neighbor of Router A and has similar configurations.
<RouterA> debugging ospf non-stop-routing
<RouterA> system-view
[RouterA] ospf non-stop-routing
// NSR backed up configuration data.
*Apr 19 10:39:17:641 2010 RouterA RM/6/RMDEBUG: NSR: Interface data of instance 0 backs up in batches
*Apr 19 10:39:17:641 2010 RouterA RM/6/RMDEBUG: NSR: Data of interface 6 backs up in batches
*Apr 19 10:39:17:641 2010 RouterA RM/6/RMDEBUG: NSR: Data of interface 5 backs up in batches
// NSR is backing up interface data in batches.
*Apr 19 10:39:17:766 2010 RouterA RM/6/RMDEBUG: NSR: backup in batches neighbor 0x3030303 of interface 152502416 in area 0 of process 1
*Apr 19 10:39:17:766 2010 RouterA RM/6/RMDEBUG: NSR: backup in batches neighbor 0x3030303 of interface 152503504 in area 1 of process 1
// NSR is backing up neighbor data in batches.
*Apr 19 10:39:17:828 2010 RouterA RM/6/RMDEBUG: NSR: Back up LSDB of instance 0 in batches
*Apr 19 10:39:17:844 2010 RouterA RM/6/RMDEBUG: NSR: Back up retransmission list of instance 0 in batches
// NSR is backing up LSDB of the process in batches.
*Apr 19 10:54:48:969 2010 RouterA RM/6/RMDEBUG: NSR: Back up configuration change of process 1 in real time
// NSR is backing up changed configuration data of the process in real time.
*Apr 19 10:53:10:172 2010 RouterA RM/6/RMDEBUG: NSR: Back up configuration of area 1 in real time
// NSR is backing up changed configuration data of the area in real time.
*Apr 19 10:45:03:110 2010 RouterA RM/6/RMDEBUG: NSR: Data of interface 5 backs up in real time enable-state of interface 5 in the area 0
// NSR is backing up interface data in real time.
*Apr 19 10:57:32:735 2010 RouterA RM/6/RMDEBUG: NSR: Back up in real time neighbor 0x3030303(7) of interface 6 in area 1 of process 1
// NSR is backing up neighbor data in real time.
*Apr 19 10:57:34:422 2010 RouterA RM/6/RMDEBUG: NSR: Back up in real time neighbor 0x3030303 data change of interface 6 in area 1 of process 1
// NSR is backing up changed neighbor data in real time.
*Apr 19 10:45:03:110 2010 RouterA RM/6/RMDEBUG: NSR: Back up in real time removal neighbor 0x3030303 of interface 5 in area 0 of process 1
// NSR is backing up in real time the neighbor to be removed.
*Apr 19 10:45:03:110 2010 RouterA RM/6/RMDEBUG: NSR: Back up retransmission node in real time: type:0x3, advertising router:0x301010b, link ID:0x3010100
// NSR is backing up the retransmission node in real time.
*Apr 19 11:03:29:703 2010 RouterA RM/6/RMDEBUG: NSR: Back up LSAs in real time type: 0x1, advertising router: 0x3030303, link ID: 0x3030303, age: 0x1, sequence number:0x8000000a, recvif:6,recvNbr:0x4010121
debugging ospf packet

Use `debugging ospf packet` to enable OSPF packet debugging.
Use `undo debugging ospf packet` to disable OSPF packet debugging.

Syntax

```
debugging ospf [ process-id ] packet [ ack | dd | hello | request | update ] [ filter { destination | source } { acl-number | prefix ip-prefix-name } ]
undo debugging ospf packet [ ack | dd | hello | request | update ] [ filter { nbr | src } { acl-number | prefix ip-prefix-name } ]
```

Default

OSPF packet debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `process-id`: Specifies an OSPF process by its ID in the range of 1 to 65535.
- `ack`: Specifies debugging for LS Ack packets.
- `dd`: Specifies debugging for Database Description (DD) packets.
- `hello`: Specifies debugging for hello packets.
- `request`: Specifies debugging for LSR packets.
- `update`: Specifies debugging for LSU packets.
- `filter`: Specifies a filtering policy.
- `destination`: Applies the filtering policy based on the destination IP address of the packets.
- `source`: Applies the filtering policy based on the source IP address of the packets.
- `acl-number`: Specifies an ACL by its number in the range of 2000 to 3999.
- `ip-prefix-name`: Specifies an IP prefix list by its name, a string of 1 to 19 characters.

Usage guidelines

If no process ID is specified, the packet debugging information of all the OSPF processes will be displayed.
Table 6 describes output fields and messages for the **debugging ospf packet** command.

### Table 108 Output from the debugging ospf packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF process-id</td>
<td>OSPF process ID.</td>
</tr>
<tr>
<td>SEND Packet</td>
<td>OSPF sent packets.</td>
</tr>
<tr>
<td>RECV Packet</td>
<td>OSPF received packets.</td>
</tr>
<tr>
<td>Source Address: src-addr</td>
<td>Source IP address of the OSPF packet.</td>
</tr>
<tr>
<td>Destination Address: dst-addr</td>
<td>Destination IP address of the OSPF packet.</td>
</tr>
</tbody>
</table>

- **Ver# ver, Type: pkt-type, Length: pkt-len**
  - OSPF packet header information:
    - ver—OSPF version 2.
    - pkt-type—OSPF packet type:
      - 1—Hello.
      - 2—DD.
      - 3—LSR.
      - 4—LSU.
      - 5—LSAck.
    - pkt-len—OSPF packet length.

- **Router: rt-id, Area: area-id, Chksum: chksum**
  - OSPF packet header information:
    - rt-id—ID of the advertising router.
    - area-id—Area ID of the sending interface.
    - chksum—Checksum of the entire packet starting from the OSPF header, excluding the 64-bit authentication field.

- **AuType: auth-type, Key(ascii): key**
  - OSPF packet header information:
    - au-type—OSPF packet authentication type:
      - 00—Non-authentication.
      - 01—Simple authentication.
      - 02—MD5 authentication.
    - key—Authentication key.

- **Net Mask: net-mask, Hello Int: hello-interval, Option: opt**
  - OSPF hello packet information:
    - net-mask—Network mask of the sending interface.
    - hello-interval—Hello interval, in seconds.
    - opt—Optional capabilities supported by the router:
      - E bit means external route support.
      - The "N" and "P" in N/P bit mean NSSA capability and Type 7 to Type 5 translation, respectively.
      - L bit means the packet carries GR-related extended data.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Rtr Priority: rt-pri, Dead Int: dead-interval, DR: ip-addr, BDR: ip-addr | OSPF hello packet information:  
  - rt-pri—Router priority.  
  - dead-interval—Neighbor dead interval, in seconds.  
  - ip-addr—IP address of the DR or BDR on the interface network segment. |
| Attached Neighbor: rt-id      | OSPF hello packet information.  
  rt-id indicates the router ID of the neighbor discovered by OSPF. |
| Hello: hello timer mismatch   | OSPF hello packet information.  
  Hello intervals of the router and the neighbor are different. |
| Hello: dead timer mismatch    | OSPF hello packet information.  
  Dead intervals of the router and the neighbor are different. |
| Hello: netmask mismatch       | OSPF hello packet information.  
  Network masks of the router and the neighbor are different. |
| Hello: extern option mismatch | OSPF hello packet information.  
  The optional capability support of the router and that of the neighbor are different. |
| Extended Options(LLS Data): opt | OSPF hello and DD packet information.  
  opt indicates the GR-related options:  
  - LR—OOB negotiation.  
  - RS—Notifying the neighbor to enter the GR (Graceful Restart) state. |
| MTU:mtu-val, Option: opt, R_I_M_MS Bit: bits | OSPF DD packet information. This argument can be a combination of the following values:  
  - mtu-val—MTU on the interface, in bytes. It is 0 if the MTU of the DD packets is not configured as that of the sending interface.  
  - opt—Optional capabilities supported by the router:  
    - E bit means external route support.  
    - The “N” and “P” in N/P bit mean NSSA capability and Type 7 to Type 5 translation, respectively.  
    - L bit means the packet carries GR-related extended data.  
  - bits—DD packet negotiation bit:  
    - I bit means the negotiation starts.  
    - M bit means more DD packets are to be exchanged.  
    - MS bit means it is the master.  
    - R bit means OOB negotiation begins. |
### Field | Description
--- | ---
DD SeqNumber: seq-num | OSPF DD packet information. seq-num indicates the DD packet sequence number.

OSPF DD, LSR, and LSAck packet information. Contents of the LSAs described in the OSPF packets:
- **ls-type** — LSA type:
  - 1 for Router LSA.
  - 2 for network LSA.
  - 3 for net-summary LSA.
  - 4 for ASBR-summary LSA.
  - 5 for AS-external–LSA.
  - 7 for NSSA LSA.
  - 9, 10, and 11 for Opaque LSA.
- **ls-id** — LSA link ID.
- **rt-id** — ID of the advertising router.

LSAType: ls-type, LinkStateId: ls-id, Advertising Rtr: rt-id

LSA Age: ls-age, Options: ExRouting:ON/OFF | OSPF DD and LSAck packet information:
- **ls-age** — LSA age.
- **ON/OFF** — Indicates whether the router supports external routing.

Length: ls-len, Seq# seq-num, CheckSum: chksum | OSPF DD and LSAck packet information:
- **ls-len** — LSA packet length.
- **seq-num** — LSA sequence number.
- **chksum** — LSA checksum.

LSACount: ls-count | OSPF LSU packet information.
*ls-count* indicates the number of LSAs in the LSU packet.

### Examples

# Enable OSPF hello packet debugging on Router A. The output in this example was created when the following conditions exist:
- Ethernet 1/0 (150.1.1.1/24) on Router A is connected to Ethernet 1/0 (150.1.1.2/24) on Router B over a broadcast network.
- On Router A, OSPF process 1 is created and area 0 is created in OSPF process 1. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
- On Router B, OSPF process 1 is created. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.

```bash
<RouterA> debugging ospf packet hello
<RouterA>
*0.68908828 RouterA RM/6/RMDEBUG:OSPF 1: SEND Packet.
*0.68908828 RouterA RM/6/RMDEBUG:Source Address: 150.1.1.1
*0.68908828 RouterA RM/6/RMDEBUG:Destination Address: 224.0.0.5
*0.68908828 RouterA RM/6/RMDEBUG:Ver# 2, Type: 1, Length: 44.
*0.68908828 RouterA RM/6/RMDEBUG:Router: 201.1.1.1, Area: 0.0.0.0, Chksum: 39833.
*0.68908828 RouterA RM/6/RMDEBUG:Autype: 00, Key(ascii): 0 0 0 0 0 0 0 0.
*0.68908828 RouterA RM/6/RMDEBUG:Net Mask: 255.255.255.0, Hello Int: 10, Option: _E_.
```
0.68908828 RouterA RM/6/RMDEBUG:Rtr Priority: 1, Dead Int: 40, DR: 150.1.1.1, BDR: 0.0.0.0.

// OSPF process 1 sent a hello packet. It has not found any neighbors.
0.68913955 RouterA RM/6/RMDEBUG:OSPF 1: RECV Packet.
0.68913955 RouterA RM/6/RMDEBUG:Source Address: 150.1.1.2
0.68913965 RouterA RM/6/RMDEBUG:Destination Address: 224.0.0.5
0.68913965 RouterA RM/6/RMDEBUG:Ver# 2, Type: 1, Length: 44.
0.68913965 RouterA RM/6/RMDEBUG:Router: 202.1.1.1, Area: 0.0.0.0, Chksum: 12700.
0.68913965 RouterA RM/6/RMDEBUG:AuType: 00, Key(ascii): 0 0 0 0 0 0 0 0.
0.68913965 RouterA RM/6/RMDEBUG:Net Mask: 255.255.255.0, Hello Int: 10, Option: _E_.
0.68913965 RouterA RM/6/RMDEBUG:Rtr Priority: 1, Dead Int: 40, DR: 0.0.0.0, BDR: 0.0.0.0.

// OSPF process 1 received a hello packet from the peer end. The peer end had not found any neighbors.
0.68918832 RouterA RM/6/RMDEBUG:OSPF 1: SEND Packet.
0.68918832 RouterA RM/6/RMDEBUG:Source Address: 150.1.1.1
0.68918842 RouterA RM/6/RMDEBUG:Destination Address: 224.0.0.5
0.68918842 RouterA RM/6/RMDEBUG:Ver# 2, Type: 1, Length: 48.
0.68918842 RouterA RM/6/RMDEBUG:Router: 201.1.1.1, Area: 0.0.0.0, Chksum: 53394.
0.68918842 RouterA RM/6/RMDEBUG:AuType: 00, Key(ascii): 0 0 0 0 0 0 0 0.
0.68918842 RouterA RM/6/RMDEBUG:Net Mask: 255.255.255.0, Hello Int: 10, Option: _E_.
0.68918852 RouterA RM/6/RMDEBUG:Rtr Priority: 1, Dead Int: 40, DR: 150.1.1.1, BDR: 0.0.0.0.
0.68918852 RouterA RM/6/RMDEBUG:Attached Neighbor: 202.1.1.1.

// OSPF process 1 sent a hello packet, which indicates the neighbor 202.1.1.1 has been found.
0.68924260 RouterA RM/6/RMDEBUG:OSPF 1: RECV Packet.
0.68924260 RouterA RM/6/RMDEBUG:Source Address: 150.1.1.2
0.68924270 RouterA RM/6/RMDEBUG:Destination Address: 224.0.0.5
0.68924270 RouterA RM/6/RMDEBUG:Ver# 2, Type: 1, Length: 48.
0.68924270 RouterA RM/6/RMDEBUG:Router: 202.1.1.1, Area: 0.0.0.0, Chksum: 14735.
0.68924280 RouterA RM/6/RMDEBUG:AuType: 00, Key(ascii): 0 0 0 0 0 0 0 0.
0.68924280 RouterA RM/6/RMDEBUG:Net Mask: 255.255.255.0, Hello Int: 10, Option: _E_.
0.68924280 RouterA RM/6/RMDEBUG:Rtr Priority: 1, Dead Int: 40, DR: 150.1.1.1, BDR: 150.1.1.2.
0.68924280 RouterA RM/6/RMDEBUG:Attached Neighbor: 201.1.1.1.

// OSPF process 1 received a hello packet from the peer end. 150.1.1.1 and 150.1.1.2 are elected as DR and BDR, respectively.
0.68928827 RouterA RM/6/RMDEBUG:OSPF 1: SEND Packet.
0.68928827 RouterA RM/6/RMDEBUG:Source Address: 150.1.1.1
0.68928827 RouterA RM/6/RMDEBUG:Destination Address: 224.0.0.5
0.68928837 RouterA RM/6/RMDEBUG:Ver# 2, Type: 1, Length: 48.
0.68928837 RouterA RM/6/RMDEBUG:Router: 201.1.1.1, Area: 0.0.0.0, Chksum: 14735.
0.68928837 RouterA RM/6/RMDEBUG:AuType: 00, Key(ascii): 0 0 0 0 0 0 0 0.
0.68928837 RouterA RM/6/RMDEBUG:Net Mask: 255.255.255.0, Hello Int: 10, Option: _E_.
0.68928847 RouterA RM/6/RMDEBUG:Rtr Priority: 1, Dead Int: 40, DR: 150.1.1.1, BDR: 150.1.1.2.
0.68928847 RouterA RM/6/RMDEBUG:Attached Neighbor: 201.1.1.1.

// OSPF process 1 sent a hello packet to maintain the neighbor relationship.
debugging ospf spf

Use **debugging ospf spf** to enable OSPF SPF debugging.

Use **undo debugging ospf spf** to disable OSPF SPF debugging.

**Syntax**

```
debugging ospf [ process-id ] spf { all | brief | intra | { asbr-summary | ase | net-summary | nssa } [ filter { acl acl-number | ip-prefix ip-prefix-name } ] }
```

```
undo debugging ospf [ process-id ] spf { all | asbr-summary | ase | brief | intra | net-summary | nssa } [ filter { acl acl-number | ip-prefix ip-prefix-name } ]
```

**Default**

OSPF SPF debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **process-id**: Specifies an OSPF process by its ID in the range of 1 to 65535.
- **all**: Specifies all the SPF scheduling and calculation debugging.
- **brief**: Specifies SPF job scheduling debugging.
- **intra**: Specifies SPF debugging for intra-area LSAs.
- **asbr-summary**: Specifies SPF debugging for ASBR-Summary LSAs.
- **ase**: Specifies SPF debugging for ASE LSAs.
- **net-summary**: Specifies SPF debugging for inter-area LSAs.
- **nssa**: Specifies SPF debugging for NSSA LSAs.
- **filter**: Filters the SPF debugging output information.
- **acl acl-number**: Specifies a basic ACL number, in the range of 2000 to 2999.
- **ip-prefix ip-prefix-name**: Specifies an IP prefix list for the filtering. It is a string of 1 to 19 characters.

**Usage guidelines**

If no process ID is specified, the SPF debugging information of all the OSPF processes will be displayed.
Table 7 describes output fields and messages for the **debugging ospf spf brief** command.

### Table 109 Output from the debugging ospf spf brief command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OSPF process-id</strong></td>
<td>OSPF process ID.</td>
</tr>
<tr>
<td>Schedule Event: schedule-event at X ms</td>
<td>SPF schedule event. It can be 0x80000000, 0x40000000, 0x10000000, 0x08000000, 0x00008000, 0x00004000, 0x00002000, 0x00001000, 0x00000800, 0x00000400, 0x00000200, or 0x00000100.</td>
</tr>
<tr>
<td>Schedule Flag: schedule-flag SPF is Scheduled</td>
<td>SPF schedule flag indicating SPF is scheduled. schedule-flag: Schedule flag. It can be 0x80000000, 0x40000000, 0x20000000, 0x10000000, 0x08000000, 0x00008000, 0x00004000, 0x00002000, 0x00001000, 0x00000800, 0x00000400, 0x00000200, or 0x00000100.</td>
</tr>
<tr>
<td>Schedule Flag: schedule-flag SPF is Stopped</td>
<td>Schedule flag indicating SPF is stopped. schedule-flag: Schedule flag. It can be 0x80000000, 0x40000000, 0x20000000, 0x10000000, 0x08000000, 0x00008000, 0x00004000, 0x00002000, 0x00001000, 0x00000800, 0x00000400, 0x00000200, or 0x00000100.</td>
</tr>
<tr>
<td>Pre Proc: Schedule: schedule-flag</td>
<td>Present SPF schedule flag. schedule-flag: Schedule flag. It can be 0x80000000, 0x40000000, 0x20000000, 0x10000000, 0x08000000, 0x00008000, 0x00004000, 0x00002000, 0x00001000, 0x00000800, 0x00000400, 0x00000200, 0x00000100, 0x00000080, 0x00000040, 0x00000020, or 0x00000010.</td>
</tr>
<tr>
<td>Pre Proc: Running: running-flag</td>
<td>Present SPF calculation flag. running-flag: Running flag. It can be 0x80000000, 0x40000000, 0x20000000, 0x10000000, 0x08000000, 0x00008000, 0x00004000, 0x00002000, 0x00001000, 0x00000800, 0x00000400, 0x00000200, 0x00000100, 0x00000080, 0x00000040, 0x00000020, 0x00000008, or 0x00000004. It can also be a combination of these values.</td>
</tr>
<tr>
<td>FRR SPF Calc On Nbr Begin</td>
<td>In FRR auto-mode, an SPF route computation began with the root as a neighboring node.</td>
</tr>
<tr>
<td>FRR SPF Calc On Nbr End</td>
<td>In FRR auto-mode, an SPF route computation ended with the root as a neighboring node.</td>
</tr>
<tr>
<td>FRR Inter-Area ASBR Route Calc On Nbr Begin</td>
<td>In FRR auto-mode, an inter-area ASBR route computation began with the root as a neighboring node.</td>
</tr>
<tr>
<td>FRR Inter-Area ASBR Route Calc On Nbr End</td>
<td>In FRR auto-mode, an inter-area ASBR route computation ended with the root as a neighboring node.</td>
</tr>
<tr>
<td>FRR Intra-Area RtrRoute BkNextHop Calc Begin</td>
<td>In FRR auto-mode, the calculation for the backup next hop of intra-area routes of the router type began.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
FRR Intra-Area RtrRoute BkNextHop Calc End | In FRR auto-mode, the calculation for the backup next hop of intra-area routes of the router type ended.
FRR Inter-Area RtrRoute BkNextHop Calc Begin | In FRR auto-mode, the calculation for the backup next hop of inter-area routes of the router type began.
FRR Inter-Area RtrRoute BkNextHop Calc End | In FRR auto-mode, the calculation for the backup next hop of inter-area routes of the router type ended.
FRR Intra-AS NetRoute BkNextHop Calc Begin | In FRR auto-mode, the calculation for the backup next hop of intra-area and inter-area routes of the network type began.
FRR Intra-AS NetRoute BkNextHop Calc End | In FRR auto-mode, the calculation for the backup next hop of intra-area and inter-area routes of the network type ended.
FRR Inter-AS NetRoute BkNextHop Calc Begin | In FRR auto-mode, the calculation for the backup next hop of routes of the ASE and NSSA types began.
FRR Inter-AS NetRoute BkNextHop Calc End | In FRR auto-mode, the calculation for the backup next hop of routes of the ASE and NSSA types ended.
ISPF-CALC: Area area-id ISPF Calculate Start at time ms | ISPF computation in the area starts.
• area-id—Area ID.
• time—Start time.
ISPF-CALC: Area area-id ISPF Calculate End at time ms | ISPF computation in the area ends.
• area-id—Area ID.
• time—End time.
ISPF-CALC: Area area-id NOMAL FULL ISPF | ISPF full computation in the area was performed.
area-id indicates the area ID.
ISPF-CALC: Area area-id TE FULL ISPF | ISPF TE full computation in the area was performed.
area-id indicates the area ID.
ISPF-CALC: Area area-id SUBTREE ISPF | ISPF subtree computation in the area was performed.
area-id indicates the area ID.
ISPF-CALC: Area area-id INIT ISPF | ISPF initialization in the area was performed.
area-id indicates the area ID.
ISPF-CALC: Area area-id UNINIT ISPF | ISPF deinitialization in the area was performed.
area-id indicates the area ID.
ISPF-CALC: Area area-id ISPF Topo Detect | ISPF topology detection in the area was performed.
area-id indicates the area ID.

Table 8 describes output fields and messages for the **debugging ospf spf intra** command.

**Table 110 Output from the debugging ospf spf intra command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF process-id</td>
<td>OSPF process ID.</td>
</tr>
<tr>
<td>Process (Intra) Area area-id</td>
<td>SPF calculation in area area-id started. area-id indicates the area ID.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Add root to area area-id Candidate List</td>
<td>OSPF added the root node to the candidate list of SPF calculation in area area-id.</td>
</tr>
</tbody>
</table>
| Rtr-LSA link-id, adv rt-id, link count link-count | Information of the router LSA associated with candidate nodes:  
- link-id—Link ID of the router LSA.  
- rt-id—ID of the advertising router.  
- link-count—Number of links in the router LSA. |
| Net-LSA link-id, adv rt-id, router count : rt-count | Information of the network LSA associated with candidate nodes:  
- link-id—Link ID of the network LSA.  
- rt-id—ID of the advertising router.  
- rt-count—Number of routers listed in the network LSA. |
| link-type Link link-id, Data link-data, cost cost | Description of each link in the router LSA:  
- link-type—Link type. It can be P-2-P, TransNet, StubNet, or Virtual.  
- link-id—Link ID.  
- link-data—Link data.  
- cost—Link cost. |
| Attach Router: router-id | Neighbor described in the network LSA. router-id indicates the ID of the router. |
| Drop for reason | Reasons for dropping a candidate node:  
- Maxage.  
- Not found neighbor.  
- Cost exceeds LSInfinity.  
- No back link.  
- Check neighbor in SPF tree fail.  
- In spf tree.  
- Next hop calculation fail.  
- No next hop.  
- Add into candidate list fail.  
- Find candidate list fail.  
- Old vertex is better. |
| Drop neighbor Rtr-Lsa link-id for reason | Reasons for dropping a candidate node (router node):  
- link-id—Link ID of the router LSA.  
- reason—Reason for dropping the node. |
| Drop neighbor Net-Lsa link-id net-mask for reason | Reasons for dropping a candidate node (network node):  
- link-id—Link ID of the network LSA.  
- net-mask—Network mask in the network LSA.  
- reason—Reason for dropping the node. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| op-type vertex:Rtr-Lsa link-id, Cost to root cost, Nh next-hop | Operation on the current candidate node (router node).  
  - **op-type**—Type of operation to perform on the candidate node. It can be Add, Update, Remove, or Get.  
  - **link-id**—Link ID of the router LSA.  
  - **cost**—Cost to the root node.  
  - **next-hop**—Next hop. |
| op-type vertex:Net-Lsa link-id net-mask, Cost to root cost, Nh next-hop | Operation on the current candidate node (network node).  
  - **op-type**—Type of operation to perform on the candidate node. It can be Add, Update, Remove, or Get.  
  - **link-id**—Link ID of the router LSA.  
  - **net-mask**—Network mask in the network LSA.  
  - **cost**—Cost to the root node.  
  - **next-hop**—Next hop. |
| Can't find old route | There is no such route in the routing table. |
| op-type path-type route dest-ip net-mask, nh next-hop, cost cost | Operation on the routing table entry.  
  - **op-type**—Type of operation to perform on the candidate node. It can be Add, Update, Remove, or Get.  
  - **path-type**—Route type. It can be unrecognized, intra-area, transit, stub, inter-area, ase, type1, or type2.  
  - **dest-ip**—Destination IP address.  
  - **next-hop**—Next hop.  
  - **cost**—Path cost. |
| Find old route for net-addr / mask-len * route-type route, nh next-hop, cost cost | The corresponding original route is found.  
  - **net-addr**—Network address.  
  - **mask-len**—Length of IP address mask.  
  - **route-type**—Route type. It can be transit or stub.  
  - **next-hop**—Next hop.  
  - **cost**—Route cost. |
| op-type route-type route, cost cost, nh next-hop | Operation on the router route.  
  - **op-type**—Operation type. It can be Old, Add, or Update.  
  - **route-type**—Route type. It can be ABR, ASBR, ABR/ASBR, or Rtr.  
  - **cost**—Path cost.  
  - **next-hop**—Next hop. |
| Stub Route: DEST: dest-addr MASK: mask-len, cost: cost | Stub route information:  
  - **dest-addr**—Destination network address.  
  - **mask-len**—Network mask length.  
  - **cost**—Route cost. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add nexthop <code>next-hop</code> to candidate nexthop list</td>
<td>In FRR auto-mode, a next hop is added to the candidate nexthop list.</td>
</tr>
<tr>
<td>Candidate nexthop <code>next-hop</code>, Nbr <code>nbr-id</code></td>
<td>In FRR auto-mode, the candidate next hop is processed.</td>
</tr>
<tr>
<td></td>
<td>• <code>next-hop</code>—Candidate next hop address.</td>
</tr>
<tr>
<td></td>
<td>• <code>nbr-id</code>—Router ID of the neighbor corresponding to the candidate next hop.</td>
</tr>
<tr>
<td>Area <code>area-id</code> Get candidate nexthop <code>next-hop</code></td>
<td>In FRR auto-mode, the candidate next hop is processed.</td>
</tr>
<tr>
<td></td>
<td>• <code>next-hop</code>—Candidate next hop address.</td>
</tr>
<tr>
<td></td>
<td>• <code>area-id</code>—ID of the area to which the primary next hop belongs.</td>
</tr>
<tr>
<td>Old BkNexthop <code>next-hop</code>, <code>attr-list</code></td>
<td>In FRR auto-mode, the attribute of the old backup next hop is listed.</td>
</tr>
<tr>
<td></td>
<td>• <code>next-hop</code>—Backup next hop address.</td>
</tr>
<tr>
<td></td>
<td>• <code>attr-list</code>—List to which the route belongs:</td>
</tr>
<tr>
<td></td>
<td>o <code>link</code>—Link protection.</td>
</tr>
<tr>
<td></td>
<td>o <code>node</code>—Node protection.</td>
</tr>
<tr>
<td></td>
<td>o <code>primary</code>—Primary next hop.</td>
</tr>
<tr>
<td>Candidate nexthop <code>next-hop</code>, <code>attr-list</code></td>
<td>In FRR auto-mode, the attribute of the current candidate next hop is listed.</td>
</tr>
<tr>
<td></td>
<td>• <code>next-hop</code>—Backup next hop address.</td>
</tr>
<tr>
<td></td>
<td>• <code>attr-list</code>—List to which the route belongs:</td>
</tr>
<tr>
<td></td>
<td>o <code>link</code>—Link protection.</td>
</tr>
<tr>
<td></td>
<td>o <code>node</code>—Node protection.</td>
</tr>
<tr>
<td></td>
<td>o <code>primary</code>—Primary next hop.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>In FRR auto-mode, the candidate next hop is selected because it is more optimal than the old backup next hop.</td>
<td>Candidate nexthop reason, update backup nexthop</td>
</tr>
<tr>
<td>The reasons why the candidate next hop is selected as the route backup next hop include the following:</td>
<td></td>
</tr>
<tr>
<td>• Is loop-free.</td>
<td></td>
</tr>
<tr>
<td>• Is primary.</td>
<td></td>
</tr>
<tr>
<td>• Is node-protect.</td>
<td></td>
</tr>
<tr>
<td>• Is link-protect.</td>
<td></td>
</tr>
<tr>
<td>• Is downstream.</td>
<td></td>
</tr>
<tr>
<td>• Is closer to destination.</td>
<td></td>
</tr>
<tr>
<td>• Has smaller IP address.</td>
<td></td>
</tr>
<tr>
<td>In FRR auto-mode, the old backup next hop is reserved because it is more optimal than the candidate next hop.</td>
<td>Candidate nexthop reason, drop candidate nexthop</td>
</tr>
<tr>
<td>The reasons why the candidate next hop is discarded include the following:</td>
<td></td>
</tr>
<tr>
<td>• Is not loop-free.</td>
<td></td>
</tr>
<tr>
<td>• Is not primary.</td>
<td></td>
</tr>
<tr>
<td>• Is not node-protect.</td>
<td></td>
</tr>
<tr>
<td>• Is not link-protect.</td>
<td></td>
</tr>
<tr>
<td>• Is not downstream.</td>
<td></td>
</tr>
<tr>
<td>• Is not closer to destination.</td>
<td></td>
</tr>
<tr>
<td>• Has larger IP address.</td>
<td></td>
</tr>
</tbody>
</table>

In FRR auto-mode, the result of calculating the backup next hop for intra-area routes of the router type is listed.

- **router-route-type** — Router type, which can be ABR, ASBR, ABR/ASBR, or Rtr.
- **dest-id** — Destination IP address.
- **next-hop** — Next hop address.
- **backup-next-hop** — Backup next hop address.
- **attr-list** — List to which the route belongs. The value can be link (link protection), node (node protection), or primary (primary next hop).

In FRR auto-mode, the intra-area routes of the network type inherit the backup next hop of the router-type routes of the corresponding advertiser.

- **net-route-type** — Network type, which can be Transit or Stub.
- **dest-id** — Destination IP address.
- **mask-len** — Mask length.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| net-route-type, dest-id / mask-len, nh next-hop, bnh backup-next-hop, attr-list | In FRR auto-mode, the result of calculating the backup next hop for intra-area routes of the network type is listed.  
- net-route-type—Network type, which can be Transit or Stub.  
- dest-id—Destination IP address.  
- mask-len—Mask length.  
- next-hop—Next hop address.  
- backup-next-hop—Backup next hop address.  
- attr-list—List to which the route belongs. The value can be link (link protection), node (node protection), or primary (primary next hop). |

**ISPF-VERT:** Drop for Add into Candidate List Fail  
Node dropped due to failure in adding into the candidate list.

**ISPF-VERT:** Drop for TE link in normal calc  
Node dropped due to full computation for a TE link.

**ISPF-VERT:** Drop for TENT: nbr node not found  
Node dropped due to neighbor node not found.

**ISPF-NODE:** Type: Type_id, Adv:Adv_id, Ls ID:Ls_ID  
SPF node type, advertising router, and LS ID.

**ISPF-NODE:** Process CandList Node  
Process candidate SPF node.

**ISPF-NODE:** Calc the child node  
Compute the child SPF node.

**ISPF-VERT:** action vertex:Rtr-node node_id, Cost to root cost, Nexthop Count:count, Nh nexthop  
Current router LSA node being processed.  
- action—Add, update, remove, or get.  
- node_id—Node ID.  
- cost—Node cost.  
- count—Number of next hops.  
- nexthop—Next hop address.

**ISPF-VERT:** action vertex:Net-node node_id/mask, Cost to root cost, Nexthop Count count, Nh nexthop  
Current network LSA node being processed.  
- action—Add, update, remove, or get.  
- node_id/mask—Node ID/mask.  
- cost—Node cost.  
- count—Number of next hops.  
- nexthop—Next hop address.

Table 9 describes output fields and messages for the **debugging ospf spf ase** command.

Table 111 Output from the debugging ospf spf ase command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF process-id</td>
<td>OSPF process ID.</td>
</tr>
<tr>
<td>Total route number less than maximum, the spf will be rescheduled</td>
<td>Some routes were deleted to reduce the total route number to be below the upper limit. SPF is now rescheduled.</td>
</tr>
<tr>
<td>Process (Ase)</td>
<td>ASE SPF calculation started.</td>
</tr>
<tr>
<td>Start (Incr ASE)</td>
<td>Incremental ASE calculation started.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>FULL SPF ASE Begin at bucket <code>bucket-num</code></td>
<td>Full ASE SPF calculation began at bucket <code>bucket-num</code> in the Hash table.</td>
</tr>
<tr>
<td>FULL SPF ASE Stop at bucket <code>bucket-num</code></td>
<td>Full ASE SPF calculation stopped at bucket <code>bucket-num</code> in the Hash table.</td>
</tr>
<tr>
<td>INCR SPF ASE Begin</td>
<td>Incremental ASE SPF began.</td>
</tr>
<tr>
<td>INCR SPF ASE End</td>
<td>Incremental ASE SPF ended.</td>
</tr>
<tr>
<td>LSA ID inconsistent with netmas. LSID: <code>ls-id</code>, Netmask: <code>net-mask</code></td>
<td>ASE LSA ID and network mask are inconsistent.</td>
</tr>
<tr>
<td>MaxAge and No associated route, flush and delete ASE LSA</td>
<td>Flush or delete the ASE LSAs that are aged out or have no associated routes.</td>
</tr>
<tr>
<td>ASE LSA cost is infinity, don’t calculate</td>
<td>ASE LSA cost is infinite. It will not be calculated.</td>
</tr>
</tbody>
</table>

**ASE net route, `dest-id` / `mask-len`, inherit backup next-hop**

In FRR auto-mode, the routes of the ASE type inherit the backup next hop of the following routes:
- Router type routes of the corresponding advertiser.
- Network type routes corresponding to the forwarding address.

Parameter description:
- `dest-id`—Destination IP address.
- `mask-len`—Mask length.

**ASE net route, `dest-id` / `mask-len`, nh `next-hop`, bnh `backup-next-hop`, `attr-list`**

In FRR auto-mode, the result of calculating the backup next hop for routes of the ASE type is listed.
- `dest-id`—Destination IP address.
- `mask-len`—Mask length.
- `next-hop`—Next hop address.
- `backup-next-hop`—Backup next hop address.
- `attr-list`—List to which the route belongs. The value can be link (link protection), node (node protection), or primary (primary next hop).

### Examples

# Enable debugging for OSPF job scheduling. The output in this example was created when the following conditions exist:

- Ethernet1/0 (3.3.3.2/24) on Router B is connected to Ethernet1/0 (3.3.3.1/24) on Router A over a broadcast network.
- Ethernet1/1 (4.4.4.1/24) on Router B is connected to Ethernet1/0 (4.4.4.2/24) on Router C over a broadcast network.
- OSPF process 1 is created on Router B. Area 0 and area 1 are created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 0. OSPF is enabled on Ethernet 1/1, which is configured to belong to area 1.
- OSPF process 1 is created on Router A. Area 0 is created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 0.
- OSPF process 1 is created on Router C. Area 1 is created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 1.
- On Router A, enable FRR to automatically calculate a backup next hop.
When you restart OSPF process 1, output similar to the following example is generated:

```
<RouterA>debugging ospf spf brief
<RouterA>reset ospf 1 process
Warning : Reset OSPF process? [Y/N]:
OSPF 1 Schedule Event: 0x80000000 at 1200641 ms.

// Event triggering the SPF schedule.
OSPF 1 SPF Intra Area Full Schedule

// Full SPF calculation schedule.
OSPF 1 Schedule Flag : 0x80000000 SPF is scheduled.

// SPF schedule flag.
OSPF 1 Schedule Event: 0x80000000 at 1200641 ms.
OSPF 1 SPF Intra Area Full Schedule
OSPF 1 Schedule Flag : 0x80000000 SPF is scheduled.
OSPF 1 Schedule Event: 0x00008000 at 1201422 ms.
OSPF 1 Schedule Event: 0x10000000 at 1201422 ms.
OSPF 1 Schedule Flag : 0x80000000 SPF is scheduled.
OSPF 1 Remove Unreachable Rts

// OSPF is deleting unreachable routes.
OSPF 1 Schedule Event: 0x80000000 at 1203750 ms.
OSPF 1 SPF Intra Area Full Schedule
OSPF 1 Schedule Flag : 0x80000000 SPF is scheduled.
OSPF 1 Pre Proc : Schedule: 0x80000000.

// Current SPF schedule flag.
OSPF 1 Pre Proc : Running : 0xF8000084.

// Current SPF calculation flag.
OSPF 1 ***************SPF Start (Full) at 1205625 ms***************

// Full SPF calculation.
OSPF 1 **** Process (Intra) Area 0.0.0.0. ****

// OSPF is processing intra-area SPF calculation.
OSPF 1 SPF running Inter area Begin

// Inter-area SPF calculation began.
OSPF 1 SPF running Inter area end

// Inter-area SPF calculation ended.
OSPF 1 SPF running Inter area Begin
OSPF 1 SPF running Inter area end
OSPF 1 Remove Unreachable Rts
OSPF 1 SPF running ASE Begin

// ASE SPF calculation began.
OSPF 1 SPF running ASE end

// ASE SPF calculation ended.
OSPF 1 SPF deletes Inter AS unreachable routes Begin

// OSPF began deleting inter-AS unreachable routes.
OSPF 1 Remove Unreachable Rts
```
// OSPF is deleting unreachable routes.
OSPF 1 SPF deletes Inter AS unreachable routes end

// OSPF finished deleting inter-AS unreachable routes.
OSPF 1 **** FRR SPF Calc On Nbr Begin ****
// SPF route computation began. The root is the neighboring node.
OSPF 1 **** Process (Intra) Area 0.0.0.0. ****
// OSPF is processing SPF route computation in area 0. The root is the neighboring node.
OSPF 1 **** FRR SPF Calc On Nbr End ****
// SPF route computation ended. The root is the neighboring node.
OSPF 1 **** FRR Inter-Area ASBR Route Calc On Nbr Begin ****
// Inter-area ASBR route computation began. The root is the neighboring node.
OSPF 1 **** FRR Inter-Area ASBR Route Calc On Nbr End ****
// Inter-area ASBR route computation ended. The root is the neighboring node.
OSPF 1 **** FRR Intra-Area RtrRoute BkNextHop Calc Begin ****
// Calculation for the backup next hop of intra-area routes of the router type began.
OSPF 1 **** FRR Intra-Area RtrRoute BkNextHop Calc End ****
// Calculation for the backup next hop of intra-area routes of the router type ended.
OSPF 1 **** FRR Intra-Area RtrRoute BkNextHop Calc Begin ****
// Calculation for the backup next hop of inter-area routes of the router type began.
OSPF 1 **** FRR Intra-Area RtrRoute BkNextHop Calc End ****
// Calculation for the backup next hop of inter-area routes of the router type ended.
OSPF 1 **** FRR Intra-AS NetRoute BkNextHop Calc Begin ****
// Calculation for the backup next hop of intra-area and inter-area routes of the network type began.
OSPF 1 **** FRR Intra-AS NetRoute BkNextHop Calc End ****
// Calculation for the backup next hop of intra-area and inter-area routes of the network type ended.
OSPF 1 **** FRR Inter-AS NetRoute BkNextHop Calc Begin ****
// Calculation for the backup next hop of routes of the ASE and NSSA types began.
OSPF 1 **** FRR Inter-AS NetRoute BkNextHop Calc End ****
// Calculation for the backup next hop of routes of the ASE and NSSA types ended.
OSPF 1 ******************SPF End at 1205641 ms**************
// SPF calculation ended.

# Enable SPF debugging for OSPF intra-area LSAs. The output in this example was created when the following conditions exist:
- Ethernet 1/0 (150.1.1.24) on Router A is connected to Ethernet 1/0 (150.1.1.2/24) on Router B.
- On Router A, OSPF process 1 is created and area 0 is created in OSPF process 1. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
- On Router B, OSPF process 1 is created. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
- On Router A, enable FRR to automatically calculate a backup next hop.

<RouterA>debugging ospf spf intra
150.1.1.2(Ethernet1/0) from Loading to Full
OSPF 1 Add root to area 0.0.0.0 Candidate List.

// OSPF added the root node to the candidate list of SPF calculation in area area-id.
OSPF 1 Get vertex:Rtr-Lsa 1.1.1.1, Cost to root 0,Nh
OSPF 1 Rtr-LSA 1.1.1.1,adv 1.1.1.1,link count 1
OSPF 1 TransNet Link 150.1.1.2,Data 150.1.1.1,Cost 10
OSPF 1 Add vertex:Net-Lsa 150.1.1.2/24,Cost to root 10,Nh 150.1.1.1

// OSPF added a candidate node.
OSPF 1 Get vertex:Net-Lsa 150.1.1.2/24,Cost to root 10,Nh 150.1.1.1
OSPF 1 Find old route for 150.1.1.0/24DEBUG:
OSPF 1 *transit route,nh 150.1.1.1,cost 10,id 29
OSPF 1 New route is better
OSPF 1 Update transit route 150.1.1.0/24,Nh 150.1.1.1,cost 10

// OSPF updated a new route.
OSPF 1 Net-LSA 150.1.1.2,adv 2.2.2.2,router count 2
OSPF 1 Attach Router 1.1.1.1             G:
OSPF 1 Attach Router 2.2.2.2             G:
OSPF 1 Calculate the leaf network
OSPF 1 Router Type  Dest:2.2.2.2
OSPF 1 Add root to area 0.0.0.0 Candidate List.
OSPF 1 Get vertex:Rtr-Lsa 1.1.1.1, Cost to root 0,Nh
OSPF 1 Rtr-LSA 1.1.1.1,adv 1.1.1.1,link count 1
OSPF 1 TransNet Link 150.1.1.2,Data 150.1.1.1,Cost 10
OSPF 1 Add vertex:Net-Lsa 150.1.1.2/24,Cost to root 10,Nh 150.1.1.1
OSPF 1 Find old route for 150.1.1.0/24DEBUG:
OSPF 1 *transit route,nh 150.1.1.1,cost 10,id 29
OSPF 1 New route is better
OSPF 1 Update transit route 150.1.1.0/24,Nh 150.1.1.1,cost 10
OSPF 1 Net-LSA 150.1.1.2,adv 2.2.2.2,router count 2
OSPF 1 Attach Router 1.1.1.1             G:
OSPF 1 Attach Router 2.2.2.2             G:
OSPF 1 Add nexthop 150.1.1.2 to candidate nexthop list

// OSPF added the next hop to the candidate next hop list.
OSPF 1 Add vertex:Rtr-Lsa 2.2.2.2, Cost to root 10,Nh 150.1.1.2
OSPF 1 Get vertex:Rtr-Lsa 2.2.2.2, Cost to root 10,Nh 150.1.1.2
OSPF 1 Can’t find Rtr route001 RM/6/RMDEBUG:
OSPF 1 Add ABR/ASBR route,cost 10,Nh 150.1.1.1

// OSPF added an ABR/ASBR route.
OSPF 1 Rtr-LSA 2.2.2.2,adv 2.2.2.2,link count 1
OSPF 1 TransNet Link 150.1.1.2,Data 150.1.1.2,Cost 10
OSPF 1 Calculate the leaf network.

// OSPF calculated the leaf network.
OSPF 1 ** Candidate nexthop 150.1.1.2, Nbr 2.2.2.2 **
OSPF 1 Add root to area 0.0.0.0 Candidate List.
OSPF 1 Get vertex:Rtr-Lsa 2.2.2.2, Cost to root 0,Nh
OSPF 1 Rtr-LSA 2.2.2.2,adv 2.2.2.2,link count 1
OSPF 1 Add TransNet Link 150.1.1.2, Cost 1
OSPF 1 Add vertex: Net-Lsa 150.1.1.2/24, Cost to root 1, Nh
OSPF 1 Get vertex: Net-Lsa 150.1.1.2/24, Cost to root 1, Nh
OSPF 1 Add transit route 150.1.1.0/24, Nh 0.0.0.0, Cost 1
OSPF 1 Net-LSA 150.1.1.2, adv 2.2.2.2, router count 2
OSPF 1 Attach Router 1.1.1.1
OSPF 1 Add vertex: Rtr-Lsa 1.1.1.1, Cost to root 1, Nh
OSPF 1 Attach Router 2.2.2.2
OSPF 1 Drop neighbor Rtr-Lsa 2.2.2.2 for in spf tree
OSPF 1 Get vertex: Rtr-Lsa 1.1.1.1, Cost to root 1, Nh
OSPF 1 Add Rtr route 1.1.1.1, cost 1 G:
OSPF 1 Rtr-LSA 1.1.1.1, adv 1.1.1.1, link count 1
OSPF 1 TransNet Link 150.1.1.2, Data 150.1.1.1, Cost 1
OSPF 1 Drop neighbor Net-Lsa 150.1.1.2/24 for in spf tree

// OSPF is performing SPF computation on the neighbor corresponding to the candidate next hop.
OSPF 1 ** Candidate nexthop 150.1.1.2, Nbr 2.2.2.2 **
OSPF 1 ASBR route, dest 2.2.2.2, nh 150.1.1.2
OSPF 1 ASBR route, dest 2.2.2.2, nh 150.1.1.2, bnh 0.0.0.0

// OSPF is calculating the backup next hop of intra-area routes of the router type.

# Enable SPF debugging for OSPF inter-area LSAs. The output in this example was created when the following conditions exist:
- Ethernet1/0 (150.1.1.2/24) on Router B is connected to Ethernet1/0 (150.1.1.1/24) on Router A over a broadcast network.
- Ethernet1/1 (4.4.4.1/24) on Router B is connected to Ethernet1/0 (4.4.4.2/24) on Router C over a broadcast network.
- OSPF process 1 is created on Router B. Area 0 and area 1 are created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 0. OSPF is enabled on Ethernet 1/1, which is configured to belong to area 1.
- OSPF process 1 is created on Router A. Area 0 is created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 0.
- OSPF process 1 is created on Router C. Area 1 is created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 1. Secondary addresses 1.1.1.1/24, 1.1.2.1/24, 1.1.3.1/24, and 1.1.4.1/24 are configured on Ethernet 1/0.
- On Router A, enable FRR to automatically calculate the backup next hop.

<RouterA>debugging ospf spf net-summary
OSPF 1 Add root to area 0.0.0.0 Candidate List.
OSPF 1 Get vertex: Rtr-Lsa 1.1.1.1, Cost to root 0, Nh
OSPF 1 Rtr-LSA 1.1.1.1, adv 1.1.1.1, link count 1
OSPF 1 Calculate the leaf network.
OSPF 1 Stub Rt 150.1.1.0/24, Cost: 10
OSPF 1 Find old route for 150.1.1.0/24BUG:
OSPF 1 *transit route, nh 150.1.1.1, cost 10, id 29
OSPF 1 New route is better
OSPF 1 Update stub route 150.1.1.0/24, Nh 150.1.1.1, cost 10
OSPF 1 Candidate list empty, SPF Area 0.0.0.0 finish.
OSPF 1 SPF area 0.0.0.0 running network summary
OSPF 1 NetSum-LSA 4.4.4.0/24 Adv 2.2.2.2 cost 10
OSPF 1 Don't calculate for route falling into active ABR range

OSPF 1 Router Type  Dest:3.3.3.3
OSPF 1 Router Type  Dest:2.2.2.2
OSPF 1 Delete Route Dest:4.4.4.0/24 Id:42 from RM
OSPF 1     Nexthop: 150.1.1.2 (Ethernet1/0)
OSPF 1 Delete OSPF inter-area Route:4.4.4.0/24 ID:42
OSPF 1 Delete Route Dest:11.11.11.0/24 Id:43 from RM
OSPF 1     Nexthop: 150.1.1.2 (Ethernet1/0)
OSPF 1 Delete OSPF type2 Route:11.11.11.0/24 ID:43

%Dec 12 11:43:30:924 2006 RouterA RM/3/RMLOG:OSPF-NBRCHANGE: Process 1, Neighbour
150.1.1.2(Ethernet1/0) from Loading to Full
OSPF 1 Add root to area 0.0.0.0 Candidate List.
OSPF 1 Get vertex:Rtr-Lsa 1.1.1.1, Cost to root 0,Nh
OSPF 1 Rtr-LSA 1.1.1.1,adv 1.1.1.1,link count 1
OSPF 1 TransNet Link 150.1.1.2,Data 150.1.1.1,Cost 10
OSPF 1 Add vertex:Net-Lsa 150.1.1.2/24,Cost to root 10,Nh 150.1.1.1
OSPF 1 Get vertex:Net-Lsa 150.1.1.2/24,Cost to root 10,Nh 150.1.1.1
OSPF 1 Find old route for 150.1.1.0/24EBUG:
OSPF 1 *stub route,nh 150.1.1.1,cost 10,id 29
OSPF 1 New route is better
OSPF 1 Update transit route 150.1.1.0/24,Nh 150.1.1.1,cost 10
OSPF 1 Net-LSA 150.1.1.2,adv 2.2.2.2,router count 2
OSPF 1 Attach Router 1.1.1.1 G:
OSPF 1 Drop neighbor Rtr-Lsa 1.1.1.1 for no back link
OSPF 1 Attach Router 2.2.2.2 G:
OSPF 1 Drop neighbor Rtr-Lsa 2.2.2.2 for no back link
OSPF 1 Calculate the leaf network.
OSPF 1 Candidate list empty,SPF Area 0.0.0.0 finish.
OSPF 1 SPF area 0.0.0.0 running network summary
OSPF 1 NetSum-LSA 4.4.4.0/24 Adv 2.2.2.2 cost 10
OSPF 1 Don't calculate for route falling into active ABR range

OSPF 1 Add root to area 0.0.0.0 Candidate List.
OSPF 1 Get vertex:Rtr-Lsa 1.1.1.1, Cost to root 0,Nh
OSPF 1 Rtr-LSA 1.1.1.1,adv 1.1.1.1,link count 1
OSPF 1 TransNet Link 150.1.1.2,Data 150.1.1.1,Cost 10
OSPF 1 Add vertex:Net-Lsa 150.1.1.2/24,Cost to root 10,Nh 150.1.1.1
OSPF 1 Get vertex:Net-Lsa 150.1.1.2/24,Cost to root 10,Nh 150.1.1.1
OSPF 1 Find old route for 150.1.1.0/24EBUG:
OSPF 1 *transit route,nh 150.1.1.1,cost 10,id 29
OSPF 1 New route is better
OSPF 1 Update transit route 150.1.1.0/24,Nh 150.1.1.1,cost 10
OSPF 1 Net-LSA 150.1.1.2,adv 2.2.2.2,router count 2
OSPF 1 Attach Router 1.1.1.1 G:
OSPF 1 Drop neighbor Rtr-Lsa 1.1.1.1 for in spf tree
OSPF 1 Attach Router 2.2.2.2 G:
OSPF 1 Add vertex: Rtr-Lsa 2.2.2.2, Cost to root 10, Nh 150.1.1.2
OSPF 1 Get vertex: Rtr-Lsa 2.2.2.2, Cost to root 10, Nh 150.1.1.2
OSPF 1 Can't find Rtr route001 RM/6/RMDEBUG:
OSPF 1 Add ABR/ASBR route, cost 10, Nh 150.1.1.2
OSPF 1 Rtr-LSA 2.2.2.2, adv 2.2.2.2, link count 1
OSPF 1 TransNet Link 150.1.1.2, Data 150.1.1.2, Cost 10
OSPF 1 Drop neighbor Net-Lsa 150.1.1.2/24 for in spf tree
OSPF 1 Calculate the leaf network.
OSPF 1 Candidate list empty, SPF Area 0.0.0.0 finish.
OSPF 1 SPF area 0.0.0.0 running network summary
OSPF 1 NetSum-LSA 4.4.4.0/24 Adv 2.2.2.2 cost 10
OSPF 1 Don't calculate for route falling into active ABR range

OSPF 1 NetSum-LSA 4.4.4.0/24 Adv 2.2.2.2 cost 10
OSPF 1 New route is better
OSPF 1 Add new Rt 4.4.4.0/24
OSPF 1 Nexthop: 150.1.1.2 (Ethernet1/0)
OSPF 1 New route is better
OSPF 1 Add new Rt 11.11.11.0/24
OSPF 1 Nexthop: 150.1.1.2 (Ethernet1/0)

// OSPF performed net-summary calculation.
OSPF 1 Inter area net route, 1.1.1.0/24, nh 150.1.1.2
OSPF 1 Inter area net route, 1.1.4.0/24, Inherit backup nexthop
OSPF 1 Inter area net route, 1.1.4.0/24, nh 150.1.1.2, bnh 0.0.0.0
OSPF 1 Inter area net route, 4.4.4.0/24, nh 150.1.1.2
OSPF 1 Inter area net route, 4.4.4.0/24, Inherit backup nexthop
OSPF 1 Inter area net route, 4.4.4.0/24, nh 150.1.1.2, bnh 0.0.0.0
OSPF 1 Inter area net route, 1.1.1.0/24, nh 150.1.1.2
OSPF 1 Inter area net route, 1.1.1.0/24, Inherit backup nexthop
OSPF 1 Inter area net route, 1.1.1.0/24, nh 150.1.1.2, bnh 0.0.0.0
OSPF 1 Inter area net route, 1.1.2.0/24, nh 150.1.1.2
OSPF 1 Inter area net route, 1.1.2.0/24, Inherit backup nexthop
OSPF 1 Inter area net route, 1.1.2.0/24, nh 150.1.1.2, bnh 0.0.0.0
OSPF 1 Inter area net route, 1.1.3.0/24, Inherit backup nexthop
OSPF 1 Inter area net route, 1.1.3.0/24, nh 150.1.1.2, bnh 0.0.0.0

// OSPF performed inter-area route backup next hop calculation.

# Enable SPF debugging for OSPF inter-area LSAs. The output in this example was created when the following conditions exist:

- Ethernet1/0 (150.1.1.2/24) on Router B is connected to Ethernet1/0 (150.1.1.1/24) on Router A over a broadcast network.
- Ethernet1/1 (4.4.4.1/24) on Router B is connected to Ethernet1/0 (4.4.4.2/24) on Router C over a broadcast network.
- OSPF process 1 is created on Router B. Area 0 and area 1 are created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 0. OSPF is enabled on Ethernet 1/1, which is configured to belong to area 1.
- OSPF process 1 is created on Router A. Area 0 is created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 0.
- OSPF process 1 is created on Router C. Area 1 is created in OSPF process 1. OSPF is enabled on Ethernet 1/0, which is configured to belong to area 1. Static routes are configured and redistributed into OSPF process 1.
- On Router A, enable FRR to automatically calculate the backup next hop.

```
<RouterA>debugging ospf spf ase
OSPF 1 Process (Ase)
// ASE SPF calculation began.
OSPF 1 FULL SPF ASE Begin at bucket 0
OSPF 1 New route is better
OSPF 1 Process (Ase)
OSPF 1 FULL SPF ASE Begin at bucket 0
// SPF calculation started from the bucket 0 of the hash table.
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
OSPF 1 New route is better
// OSPF performed ASE SPF calculation debugging.
OSPF 1 ASE net route,100.1.9.0/24,nh 150.1.1.2
OSPF 1 ASE net route,100.1.9.0/24,Inherit backup nexthop
OSPF 1 ASE net route,100.1.9.0/24,nh 150.1.1.2,bnh 0.0.0.0
OSPF 1 ASE net route,100.1.10.0/24,nh 150.1.1.2
OSPF 1 ASE net route,100.1.10.0/24,Inherit backup nexthop
OSPF 1 ASE net route,100.1.10.0/24,nh 150.1.1.2,bnh 0.0.0.0
OSPF 1 ASE net route,100.1.11.0/24,nh 150.1.1.2
OSPF 1 ASE net route,100.1.11.0/24,Inherit backup nexthop
OSPF 1 ASE net route,100.1.11.0/24,nh 150.1.1.2,bnh 0.0.0.0
OSPF 1 ASE net route,100.1.1.0/24,nh 150.1.1.2
OSPF 1 ASE net route,100.1.1.0/24,Inherit backup nexthop
OSPF 1 ASE net route,100.1.1.0/24,nh 150.1.1.2,bnh 0.0.0.0
OSPF 1 ASE net route,100.1.2.0/24,nh 150.1.1.2
OSPF 1 ASE net route,100.1.2.0/24,Inherit backup nexthop
OSPF 1 ASE net route,100.1.2.0/24,nh 150.1.1.2,bnh 0.0.0.0
OSPF 1 ASE net route,100.1.3.0/24,nh 150.1.1.2
OSPF 1 ASE net route,100.1.3.0/24,Inherit backup nexthop
OSPF 1 ASE net route,100.1.3.0/24,nh 150.1.1.2,bnh 0.0.0.0
OSPF 1 ASE net route,100.1.4.0/24,nh 150.1.1.2
OSPF 1 ASE net route,100.1.4.0/24,Inherit backup nexthop
OSPF 1 ASE net route,100.1.4.0/24,nh 150.1.1.2,bnh 0.0.0.0
```
debugging ospf timer

Use `debugging ospf timer` to enable OSPF timer debugging.

Use `undo debugging ospf timer` to disable OSPF timer debugging.

Syntax

```
debugging ospf [ process-id ] timer [ lsa-generate | spf ]
undo debugging ospf [ process-id ] timer [ lsa-generate | spf ]
```

Default

OSPF timer debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `process-id`: Specifies an OSPF process by its ID in the range of 1 to 65535.
- `lsa-generate`: Specifies LSA generation timer debugging.
- `spf`: Specifies SPF calculation timer debugging.

Usage guidelines

If no process ID is specified, the timer debugging information of all the OSPF processes will be displayed.

Examples

# Enable OSPF timer debugging on Router A. The output in this example was created when the following conditions exist:

- Ethernet 1/0 (150.1.1.1/24) on Router A is connected to Ethernet 1/0 (150.1.1.2/24) on Router B over a broadcast network.
- On Router A, OSPF process 1 is created and area 0 is created in OSPF process 1. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
- On Router B, OSPF process 1 is created. Ethernet 1/0 is enabled with OSPF and is configured to belong to area 0.
When you restart OSPF process 1, output similar to the following example is generated:

```<RouterA>debugging ospf timer
<RouterA>reset ospf 1 process
Warning : Reset OSPF process? [Y/N]:y
%Sep 18 11:21:04:612 2006 RouterA RM/3/RMLOG:OSPF-NBRCHANGE: Process 1, Neighbour 150.1.1.2(Ethernet1/0) from Full to Down
  OSPF 1 reset SPF TIMER,timeout value is 5000 ms
  // OSPF reset the SPF calculation timer with a timeout value of 5000 milliseconds.
  OSPF 1 delete SPF TIMER R1 RM/6/RMDEBUG:
  // OSPF deleted the SPF calculation timer.
  OSPF 1 delete MIN LS TIMER R1 RM/6/RMDEBUG:
  // OSPF deleted the LSA generation timer.
  OSPF 1 create SPF TIMER,timeout value is 5000 ms
  // OSPF created the SPF calculation timer with a timeout value of 5000 milliseconds.
  OSPF 1 create MIN LS TIMER,timeout value is 5000 ms
  // OSPF created the LSA generation timer with a timeout value of 5000 milliseconds.
  OSPF 1 reset MIN LS TIMER,timeout value is 714 ms
  // OSPF reset the LSA generation timer with a timeout value of 714 milliseconds.
  OSPF 1 reset SPF TIMER,timeout value is 5000 ms
  OSPF 1 reset MIN LS TIMER,timeout value is 5000 ms
%Sep 18 11:21:13:908 2006 RouterA RM/3/RMLOG:OSPF-NBRCHANGE: Process 1, Neighbour 150.1.1.2(Ethernet1/0) from Loading to Full
  OSPF 1 reset MIN LS TIMER,timeout value is 20 ms
  OSPF 1 delete SPF TIMER RouterA RM/6/RMDEBUG:
  OSPF 1 create SPF TIMER,timeout value is 5000 ms
```
OSPFv3 debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging ospfv3 event

Use `debugging ospfv3 event` to enable OSPFv3 event debugging, such as ABR, ASBR, or virtual link event debugging.

Use `undo debugging ospfv3 event` to disable OSPFv3 event debugging.

**Syntax**

```
debugging ospfv3 event { abr | all | bfd | asbr | vlink }
undo debugging ospfv3 event { abr | all | asbr | bfd | vlink }
```

**Default**

OSPFv3 event debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **abr**: Specifies ABR event debugging.
- **all**: Specifies all event debugging.
- **asbr**: Specifies ASBR event debugging.
- **bfd**: Specifies BFD event debugging.
- **vlink**: Specifies virtual link event debugging.

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging ospfv3 event abr` command.

**Table 1: Output from the debugging ospfv3 event abr command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPFv3 ROUTER[number]</td>
<td>OSPFv3 process number.</td>
</tr>
<tr>
<td>Change status to string</td>
<td>Router status change. string indicates router status type. It can be ABR or non-ABR.</td>
</tr>
<tr>
<td>Donot creat default LSA for no FULL neighbors</td>
<td>No default Type-3 LSA is generated because there is no FULL neighbor in the backbone area.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the `debugging ospfv3 event bfd` command.
Table 113 Output from the debugging ospfv3 event bfd command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Type XXXX</td>
<td>Type of message that OSPF uses to notify information to BFD, including create session and delete session.</td>
</tr>
<tr>
<td>Connect Type direct-connect</td>
<td>The connection type of the BFD session is direct connection.</td>
</tr>
<tr>
<td>Src IP Address X::X</td>
<td>Source IPv6 address.</td>
</tr>
<tr>
<td>Dst IP Address X::X</td>
<td>Destination IPv6 address.</td>
</tr>
</tbody>
</table>

Examples

# Enable ASBR event debugging on Router A. The output in this example was created when the following conditions exist:
- Router A is connected to Router B through Ethernet 1/0.
- OSPFv3 process 1 is created on Router A. OSPFv3 is enabled on Ethernet 1/0.
- OSPFv3 process 1 is created on Router B. OSPFv3 is enabled on the corresponding interface.
- Router A and Router B are configured to belong to area 0. On Router A, static route redistribution is configured and then canceled.

<Sysname> debugging ospfv3 event asbr
*0.2383219 Sysname RM/6/RMDEBUG:OSPFv3 ROUTER: change status to non-ASBR
// The router status changed from ASBR to non-ASBR.
*0.2649563 Sysname RM/6/RMDEBUG:OSPFv3 ROUTER: change status to ASBR
// The router status changed from non-ASBR to ASBR.

debugging ospfv3 graceful-restart

Use debugging ospfv3 graceful-restart to enable OSPFv3 GR debugging.
Use undo debugging ospfv3 graceful-restart to disable OSPFv3 GR debugging.

Syntax

debugging ospfv3 graceful-restart
undo debugging ospfv3 graceful-restart

Default

OSPFv3 GR debugging is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

Table 3 describes output fields and messages for the debugging ospfv3 graceful-restart command.
Table 114 Output from the debugging ospfv3 graceful-restart command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>received smooth message</td>
<td>OSPFv3 received a GR message.</td>
</tr>
<tr>
<td>Notify RM enter GR</td>
<td>OSPFv3 notified the route management module that the device enters the GR process.</td>
</tr>
<tr>
<td>Grace-LSA sending timer expired on interface IFNAME</td>
<td>Grace-LSA sending timer expired on interface IFNAME specified by the interface type and interface number.</td>
</tr>
<tr>
<td>Received maxage grace-LSA from neighbor RouterID on interface IFNAME</td>
<td>OSPFv3 received a Grace-LSA with the age of 3600 seconds from the neighbor.</td>
</tr>
<tr>
<td>• RouterID — Router ID of the neighbor.</td>
<td>• IFNAME — Interface specified by the interface type and interface number.</td>
</tr>
<tr>
<td>Flush self stale LSAs</td>
<td>OSPFv3 deleted local stale LSAs.</td>
</tr>
<tr>
<td>Schedule whole route calculation</td>
<td>OSPFv3 triggered the calculation of all routes.</td>
</tr>
<tr>
<td>Interface IFID add rebuild neighbor NBRID in area AREAID</td>
<td>OSPFv3 re-established neighbor NBRID on the interface IFID of area AREAID.</td>
</tr>
<tr>
<td>• IFID — Interface ID.</td>
<td>• NBRID — ID of the neighbor of the interface.</td>
</tr>
<tr>
<td>• AREAID — ID of the area where the interface belongs.</td>
<td></td>
</tr>
<tr>
<td>Maxage LSAs don’t rebuild neighbor</td>
<td>The LSAs exceeding 3600 seconds cannot be used to establish neighbors.</td>
</tr>
<tr>
<td>Cancel interface IFNAME DR election for in GR</td>
<td>The DR election cannot be performed during a GR process.</td>
</tr>
<tr>
<td>The age of the grace-LSA is more than the grace period</td>
<td>The age of the Grace-LSA exceeds the GR interval.</td>
</tr>
<tr>
<td>Flush all self grace-LSAs</td>
<td>OSPFv3 deleted the Grace-LSAs generated on all the local interfaces.</td>
</tr>
<tr>
<td>Start flush stale AS-external-LSAs</td>
<td>OSPFv3 started to delete stale AS-external-LSA.</td>
</tr>
<tr>
<td>Don’t rebuild neighbor for no network-LSA of interface IFID</td>
<td>OSPFv3 failed to establish adjacencies because no Network-LSAs of interface IFID exist.</td>
</tr>
</tbody>
</table>

Examples

# Enable GR debugging on Router A. The output in this example was created when the following conditions exist:
- Router A is connected to Router B through Ethernet 1/1.
- Create OSPFv3 process 1 and enable GR for Router A to act as a GR restarter. OSPFv3 is enabled on Ethernet 1/1.
- OSPFv3 process 1 is created on Router B, which is a GR helper by default. OSPFv3 is enabled on Ethernet 1/1.
- Router A and Router B are configured to belong to area 0.
When you perform a master/backup switchover on Router A, output similar to the following example is generated:

```
<RouterA> debugging ospfv3 graceful-restart
from Full to Down
*Jun 12 11:10:14:00 2008 RouterA RM/6/RMDEBUG:OSPFv3 1 GR: Enter GR.

// Router A entered the GR process.

// OSPFv3 sent Grace-LSA to the GR helper.
*Jun 12 11:10:25:281 2008 RouterA RM/6/RMDEBUG:OSPFv3 Cancel interface Ethernet1/1 DR election for in GR
from Exchange to Full
*Jun 12 11:10:25:375 2008 RouterA RM/6/RMDEBUG:OSPFv3 1 GR: Interface 0.143.0.8 add rebuild
neighbor 8.8.8.8 in area 0.0.0.0.
*Jun 12 11:10:25:375 2008 RouterA RM/6/RMDEBUG:OSPFv3 1 GR: Interface 0.143.0.8 delete
rebuild neighbor 8.8.8.8 in area 0.0.0.0.

// The suppress hello timer expired and normal hello packets can be forwarded.
*Jun 12 11:10:25:375 2008 RouterA RM/6/RMDEBUG:OSPFv3 1 GR: All adjacencies had been rebuilt.

// OSPFv3 finished establishing adjacencies.

// OSPFv3 was deleting all local Grace-LSAs.
```

// OSPFv3 is deleting stale LSAs.


// A route calculation was triggered.


// OSPFv3 finished route calculation.

*Jun 12 11:10:29:484 2008 RouterA RM/6/RMDEBUG:OSPFv3 1 GR: RM notified all protocols left GR.


// The route management module notified all protocols to exit the GR process.


// OSPFv3 finished GR.

# Enable GR debugging on Router B.

<RouterB> debugging ospfv3 graceful-restart


%Jun 12 11:31:05:625 2008 RouterB RM/3/RMLOG:OSPFv3 Process 1 Neighbor 9.9.9.9(Ethernet1/1) from Full to ExStart

// Router B entered the helper mode.

%Jun 12 11:31:05:672 2008 RouterB RM/3/RMLOG:OSPFv3 Process 1 Neighbor 9.9.9.9(Ethernet1/1) from Exchange to Full

// OSPFv3 is re-establishing adjacencies.

*Jun 12 11:31:05:688 2008 RouterB RM/6/RMDEBUG:OSPFv3 1 GR: Received maxage grace-LSA from neighbor 9.9.9.9 on interface Ethernet1/1.

// OSPFv3 is deleting Grace-LSAs.


// Router B exited the helper mode.

debugging ospfv3 ifsm

Use **debugging ospfv3 ifsm** to enable debugging for the OSPFv3 interface state machine.

Use **undo debugging ospfv3 ifsm** to disable debugging for the OSPFv3 interface state machine.

**Syntax**

debugging ospfv3 ifsm [ event | status | timer ]

undo debugging ospfv3 ifsm [ event | status | timer ]

**Default**

Debugging for the OSPFv3 interface state machine is disabled.
Views

User view

Default command level

1: Monitor level

Parameters

event: Specifies event debugging for the interface state machine.

status: Specifies status debugging for the interface state machine.

timer: Specifies timer debugging for the interface state machine.

Usage guidelines

Table 4 describes output fields and messages for the `debugging ospfv3 ifsm` command.

Table 115 Output from the debugging ospfv3 ifsm command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPFv3 IFSM[interfacename]</td>
<td>OSPFv3 interface state machine debugging.</td>
</tr>
<tr>
<td>state (interface-event)</td>
<td>Current interface state and interface event <code>interface-event</code>.</td>
</tr>
<tr>
<td></td>
<td>Current interface state:</td>
</tr>
<tr>
<td></td>
<td>• Down—The interface is down.</td>
</tr>
<tr>
<td></td>
<td>• Loopback—The interface is a loopback interface.</td>
</tr>
<tr>
<td></td>
<td>• Waiting—The interface is in the waiting state.</td>
</tr>
<tr>
<td></td>
<td>• P-To-P —The interface is connected to a P2P network or configured as one end of a virtual link.</td>
</tr>
<tr>
<td></td>
<td>• Backup—The interface is the backup designated router.</td>
</tr>
<tr>
<td></td>
<td>• DR—The interface is the designated router.</td>
</tr>
<tr>
<td></td>
<td>Event that triggers an interface state change:</td>
</tr>
<tr>
<td></td>
<td>• InterfaceUp—The interface state is changed from Down to UP.</td>
</tr>
<tr>
<td></td>
<td>• WaitTimer—The wait timer expires.</td>
</tr>
<tr>
<td></td>
<td>• BackupSeen—Indicates whether the interface detects a BDR on the network.</td>
</tr>
<tr>
<td></td>
<td>The interface receives from a neighbor a hello packet declaring the neighbor is a BDR, or declaring the neighbor is a DR.</td>
</tr>
<tr>
<td></td>
<td>• NeighborChange—A neighbor becomes a DR or BDR, or it is no longer a DR or BDR. A DR/BDR re-election is required.</td>
</tr>
<tr>
<td></td>
<td>• LoopInd—The interface is a loopback interface.</td>
</tr>
<tr>
<td></td>
<td>• UnLoopInd—The interface is no longer a loopback interface.</td>
</tr>
<tr>
<td></td>
<td>• InterfaceDown—The interface state changes from UP to Down.</td>
</tr>
</tbody>
</table>

DR-Election[string]: DR ROUTER-ID

Current DR. The string argument can take one of the following values:

• 1st—The first DR election is performed.
• 2nd—An interface state change occurs and a DR re-election is performed.

DR-Election[string]: BDR ROUTER-ID Backup RouterID

Current BDR. The string argument can take one of the following values:

• 1st—The first DR election is performed.
• 2nd—An interface state change occurs and a DR re-election is performed.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifsm_ignore called</td>
<td>The received event is not processed.</td>
</tr>
<tr>
<td>Interface state:</td>
<td></td>
</tr>
<tr>
<td>• <strong>Down</strong>—The interface is down.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Loopback</strong>—The interface is a loopback interface.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Waiting</strong>—The interface is in the waiting state.</td>
<td></td>
</tr>
<tr>
<td>• <strong>P-To-P</strong>—The interface is connected to a P2P network or configured as one end of a virtual link.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Backup</strong>—The interface is the backup designated router.</td>
<td></td>
</tr>
<tr>
<td>• <strong>DR</strong>—The interface is the designated router.</td>
<td></td>
</tr>
</tbody>
</table>

**string timer expire**   
Timer expired. Timers include Wait timer, Hello timer, and LS ACK timer.

**Examples**

```plaintext
# Enable OSPFv3 interface state machine debugging on Router A. The output in this example was created when the following conditions exist:
• Router A is connected to Router B through Ethernet 1/0.
• OSPFv3 process 1 is created on Router A. OSPFv3 is enabled on Ethernet 1/0.
• OSPFv3 process 1 is created on Router B. OSPFv3 is enabled on Ethernet 1/0.

When a neighbor relationship is established between Router A and Router B, output similar to the following example is generated:

<!-- Sysname --> debugging ospfv3 ifsm
  *0.85734 Sysname RM/6/RMDEBUG:OSPFv3 IFSM[Ethernet1/0]: Down (InterfaceUp)
  *0.95875 Sysname RM/6/RMDEBUG:OSPFv3 IFSM[Ethernet1/0]:
  DR-Election[1st]: DR 34.1.1.1
  // The first DR election was complete. The DR is 34.1.1.1.
  *0.95890 Sysname RM/6/RMDEBUG:OSPFv3 IFSM[Ethernet1/0]:
  DR-Election[2nd]: DR 34.1.1.1
  *0.1688515 Sysname RM/6/RMDEBUG:OSPFv3 IFSM[Ethernet1/0]: Status change Backup -> DR
  // A DR re-election was carried out and the state of Ethernet 1/0 changed from Backup to DR.
  *0.3761765 Sysname RM/6/RMDEBUG:OSPFv3 IFSM[Ethernet1/0]: Hello timer expire
  // The hello timer expired.
```

**debugging ospfv3 lsa**

Use `debugging ospfv3 lsa` to enable OSPFv3 LSA debugging.

Use `undo debugging ospfv3 lsa` to disable OSPFv3 LSA debugging.

**Syntax**

```plaintext
debugging ospfv3 lsa { all | flooding | generate | install | maxage | refresh | verbose }
undo debugging ospfv3 lsa { all | flooding | generate | install | maxage | refresh | verbose }
```

**Default**

OSPFv3 LSA debugging is disabled.
Views

User view

Default command level

1: Monitor level

Parameters

all: Specifies all LSA debugging.
flooding: Specifies LSA flooding debugging.
generate: Specifies LSA generation debugging.
install: Specifies debugging of LSAs installation into the LSDB.
maxage: Specifies LSA maxage debugging.
refresh: Specifies LSA refresh debugging.
verbose: Specifies LSA detailed debugging information.

Usage guidelines

Table 5 describes output fields and messages for the debugging ospfv3 lsa flooding command.

Table 116 Output from the debugging ospfv3 lsa flooding command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Indicates the LSDB containing the LSA. It can be:</td>
</tr>
<tr>
<td>Link</td>
<td>Identified by interface type and interface number.</td>
</tr>
<tr>
<td>Area ID.</td>
<td></td>
</tr>
<tr>
<td>AS</td>
<td>The flooding range is the entire AS.</td>
</tr>
<tr>
<td>number</td>
<td>Indicates the LSA type. It can be:</td>
</tr>
<tr>
<td>0x2001</td>
<td>Router-LSAs.</td>
</tr>
<tr>
<td>0x2002</td>
<td>Network-LSAs.</td>
</tr>
<tr>
<td>0x2003</td>
<td>Inter-Area-Prefix-LSAs.</td>
</tr>
<tr>
<td>0x2004</td>
<td>Inter-Area-Router-LSAs.</td>
</tr>
<tr>
<td>0x2005</td>
<td>AS-external LSAs.</td>
</tr>
<tr>
<td>0x0008</td>
<td>Link-LSAs.</td>
</tr>
<tr>
<td>0x2009</td>
<td>Intra-Area-Prefix-LSAs.</td>
</tr>
<tr>
<td>0x000b</td>
<td>Grace-LSA.</td>
</tr>
<tr>
<td>LSID</td>
<td>Indicates the Link-state ID of the LSA.</td>
</tr>
<tr>
<td>adv_router</td>
<td>Indicates the ID of the advertising router.</td>
</tr>
<tr>
<td>*</td>
<td>Indicates that the LSA is self-originated.</td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the debugging ospfv3 lsa maxage command.
Table 117 Output from the debugging ospfv3 lsa maxage command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Flag bit:</td>
</tr>
<tr>
<td>Flags;string</td>
<td>• V—Vlink end.</td>
</tr>
<tr>
<td></td>
<td>• E—ASBR.</td>
</tr>
<tr>
<td></td>
<td>• B—ABR.</td>
</tr>
<tr>
<td></td>
<td>• W—MOSPF support.</td>
</tr>
<tr>
<td>Options</td>
<td>Router processing capabilities:</td>
</tr>
<tr>
<td></td>
<td>• V6—A value of 0 indicates that the router/link will not take part in IPv6 route calculation.</td>
</tr>
<tr>
<td>Link connected to:</td>
<td>• E—A value of 0 indicates that the flooding of AS-External-LSAs is not supported.</td>
</tr>
<tr>
<td>linktype</td>
<td>• MC—Multicast-related.</td>
</tr>
<tr>
<td></td>
<td>• N—Indicates whether it is in a Not So Stub Area (NSSA).</td>
</tr>
<tr>
<td></td>
<td>• R—Indicates whether this is an Active Router. A value of 0 indicates that the router only forwards locally originated packets.</td>
</tr>
<tr>
<td></td>
<td>• DC—Indicates whether dial-on-demand is supported.</td>
</tr>
<tr>
<td>Options</td>
<td>Link type:</td>
</tr>
<tr>
<td></td>
<td>• (null).</td>
</tr>
<tr>
<td></td>
<td>• Point-to-point.</td>
</tr>
<tr>
<td></td>
<td>• Transit Network.</td>
</tr>
<tr>
<td></td>
<td>• (Reserved).</td>
</tr>
<tr>
<td></td>
<td>• Virtual Link.</td>
</tr>
<tr>
<td>PrefixOptions</td>
<td>PrefixOptions:</td>
</tr>
<tr>
<td>Option</td>
<td>• NU—A value of 1 indicates that the prefix will not take part in IPv6 unicast route calculation.</td>
</tr>
<tr>
<td></td>
<td>• MC—A value of 1 indicates that the prefix will take part in IPv6 multicast route calculation.</td>
</tr>
<tr>
<td></td>
<td>• LA—A value of 1 indicates that the address is the local host address. The prefix length is 128.</td>
</tr>
<tr>
<td></td>
<td>• P—Indicates whether the NSSA ABR advertises the prefix to other areas.</td>
</tr>
<tr>
<td>Destination Router ID</td>
<td>ASBR router ID.</td>
</tr>
<tr>
<td>Metric Type string</td>
<td>Metric type:</td>
</tr>
<tr>
<td></td>
<td>• 1 (Comparable directly to link state metric).</td>
</tr>
<tr>
<td></td>
<td>• 2 (Larger than any link state path).</td>
</tr>
</tbody>
</table>

Examples

# Enable LSA flooding debugging on Router A. The output in this example was created when the following conditions exist:
• Router A is connected to Router B through Ethernet 1/0.
• OSPFv3 process 1 is created on Router A. OSPFv3 is enabled on Ethernet 1/0.
• OSPFv3 process 1 is created on Router B. OSPFv3 is enabled on Ethernet 1/0.
When a neighbor relationship is established between Router A and Router B, output similar to the following example is generated:

```
<Sysname> debugging ospfv3 lsa flooding
*0.14227421 Sysname RM/6/RMDEBUG: OSPFv3 LSA[Link(Ethernet1/0):Type(0x0008):0.15.0.24:34.1.1.1 *]: consider flooding through interface[Ethernet1/0]
// OSPFv3 advertised an LSA (the LSA type is 0x0008, Link State ID is 0.15.0.24, and advertising router ID is 34.1.1.1) through Ethernet 1/0.
*0.28470631 Sysname RM/6/RMDEBUG:OSPFv3 LSA[Area(0.0.0.0):Type(0x2004):0.0.0.1:78.3.3.3]: flood started
*0.284705984 Sysname RM/6/RMDEBUG:OSPFv3 LSA[Area(0.0.0.0):Type(0x2003):0.0.0.3:78.3.3.3 *]: Update param for self-originated LSA
*0.284706046 Sysname RM/6/RMDEBUG:OSPFv3 LSA[Area(0.0.0.0):Type(0x2004):0.0.0.1:78.3.3.3 *]: Process self-originated LSA
// OSPFv3 was processing self-originated LSAs.
*0.79095047 Sysname RM/6/RMDEBUG:OSPFv3 LSA[Area(0.0.0.0):Type(0x2004):0.0.0.1:78.3.3.3]: DISCARD, same instance exists in LSDB
// OSPFv3 discarded the received LSA because the same LSA already exists in the LSDB.
*0.14241000 Sysname RM/6/RMDEBUG:OSPFv3 LSA[Area(0.0.0.0):Type(0x2009):0.0.0.2:34.1.1.1]: DISCARD, LSA is MaxAge, there is no instance and none of router's neighbors are in states Exchange or Loading
// OSPFv3 discarded the received LSA because its maximum age was reached and none of the router's neighbors is in the Exchange or Loading state.
```

**debugging ospfv3 nfsm**

Use `debugging ospfv3 nfsm` to enable debugging for the OSPFv3 neighbor state machine.

Use `undo debugging ospfv3 nfsm` to disable debugging for the OSPFv3 neighbor state machine.

**Syntax**

```
diff debugging ospfv3 nfsm [ event | status | timer ]
undo debugging ospfv3 nfsm [ event | status | timer ]
```

**Default**

Debugging for the OSPFv3 neighbor state machine is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **status**: Specifies status debugging of the neighbor state machine.
- **event**: Specifies event debugging of the neighbor state machine.
- **timer**: Specifies timer debugging of the neighbor state machine.

**Usage guidelines**

Table 7 describes output fields and messages for the `debugging ospfv3 nfsm` command.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSM[RouterID-ifid]</td>
<td>Router ID and interface ID of the neighbor router.</td>
</tr>
<tr>
<td>Status change</td>
<td>Neighbor status changed from state1 to state2. The values of state1 and state2 can be:</td>
</tr>
<tr>
<td>state1-&gt;state2</td>
<td>• <strong>Down</strong>—The neighbor is down.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Attempt</strong>—A further attempt is made to contact the neighbor after not receiving anything from it.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Init</strong>—A hello packet sent by the neighbor is received but 2-way communication is not established.</td>
</tr>
<tr>
<td></td>
<td>• <strong>2-Way</strong>—2-way communication is established.</td>
</tr>
<tr>
<td></td>
<td>• <strong>ExStart</strong>—The master/subordinate negotiation is performed.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Exchange</strong>—DD packets are exchanged.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Loading</strong>—LSR packets are sent to the neighbor.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Full</strong>—An adjacency is established.</td>
</tr>
</tbody>
</table>
Neighbor state state and triggering event event.

Current neighbor state:
- **Down**—The neighbor is down.
- **Attempt**—A further attempt is made to contact the neighbor after receiving nothing from it.
- **Init**—The hello packets sent by the neighbor were received but 2-way communication is not established with the neighbor.
- **2-Way**—2-way communication is established.
- **ExStart**—The master/subordinate negotiation is performed.
- **Exchange**—DD packets are exchanged.
- **Loading**—LSR packets are sent to the neighbor.
- **Full**—An adjacency is established.

Triggering event:
- **HelloReceived**—A hello packet is received from the neighbor.
- **Start**—Hello packets will be sent to the neighbor at an interval of HelloInterval. This event is only related to a neighbor on an NBMA network.
- **2-WayReceived**—2-way communication is established between the two neighboring routers.
- **NegotiationDone**—The master/subordinate negotiation is completed and DD sequence numbers are exchanged.
- **ExchangeDone**—The two routers have exchanged DD packets successfully and know which LSAs of the neighbor are missing from their LSDBs.
- **BadLSReq**—The received LSR contains LSAs that cannot be found in the local LSDB.
- **LoadingDone**—The LSR list is empty after the database exchange is complete.
- **AdjOK**—Indicates whether an adjacency needs to be established or maintained with the neighbor.
- **SeqNumberMismatch**—An error has occurred when an adjacency is being established.
- **1-WayReceived**—The hello packet received from the neighbor does not contain the local router.
- **KillNbr**—It is impossible to communicate with the neighbor and therefore the neighbor is forced to enter the Down state.
- **InactivityTimer**—No hello packet has been received from the neighbor recently and therefore the neighbor dead timer is started.
- **LLDown**—The neighbor is unreachable and therefore is forced to enter the Down state.

The received event is not processed.

Timer expired. Timers include:
- Inactivity timer.
- DD retransmit timer.
- LS update timer.
- LS request timer.
Examples

# Enable neighbor state machine debugging on Router A. The output in this example was created when the following conditions exist:

- Router A is connected to Router B through Ethernet 1/0.
- OSPFv3 process 1 is created on Router A. OSPFv3 is enabled on Ethernet 1/0.
- OSPFv3 process 1 is created on Router B. OSPFv3 is enabled on Ethernet 1/0.

```
<Sysname> debugging ospfv3 nsf
```

*0.5752906 Sysname RM/6/RMDEBUG:OSPFv3 NFSM[34.1.1.1-000f0018]: Down (HelloReceived)

// The Router ID, interface ID, and state of the neighbor is 34.1.1.1, 000f0018, and Down, respectively. The HelloReceived event was triggered.

*0.5461109 Sysname RM/6/RMDEBUG:OSPFv3 NFSM[34.1.1.1-000f0018]: LS update timer expire

// The Router ID and interface ID of the neighbor are 34.1.1.1 and 000f0018, respectively. The update packet timer expired.

*0.4252250 Sysname RM/6/RMDEBUG:OSPFv3 NFSM[34.1.1.1-000f0018]: Status change Full -> Down

// The Router ID and interface ID of the neighbor are 34.1.1.1 and 000f0018, respectively. The neighbor state changed from Full to Down.

debugging ospfv3 packet

Use debugging ospfv3 packet to enable OSPFv3 packet debugging.

Use undo debugging ospfv3 packet to disable OSPFv3 packet debugging.

Syntax

debugging ospfv3 packet { all | { ack | dd | hello | request | update } * } [ verbose ]

undo debugging ospfv3 packet { all | { ack | dd | hello | request | update } * } [ verbose ]

Default

OSPFv3 packet debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- all: Specifies all packet debugging.
- ack: Specifies LSAck packet debugging.
- dd: Specifies DD packet debugging.
- hello: Specifies hello packet debugging.
- request: Specifies LSR packet debugging.
- update: Specifies LSU packet debugging.
- verbose: Specifies detailed packet debugging information.
Usage guidelines

Table 8 describes output fields and messages for the **debugging ospfv3 packet** command.

### Table 119 Output from the debugging ospfv3 packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Packet[event]: src(source-ipv6) -&gt; dst(dest-ipv6)</strong></td>
<td>OSPFv3 sent/received hello packets.</td>
</tr>
<tr>
<td></td>
<td>• <strong>event</strong>—RECV indicates that a hello packet is received. SEND indicates that a hello packet is sent.</td>
</tr>
<tr>
<td></td>
<td>• <strong>source-ipv6</strong>—Source IPv6 link-local address of the hello packet.</td>
</tr>
<tr>
<td></td>
<td>• <strong>dest-ipv6</strong>—Destination IPv6 link-local address of the hello packet.</td>
</tr>
<tr>
<td><strong>Type number(string)</strong></td>
<td>OSPFv3 packet type.</td>
</tr>
<tr>
<td></td>
<td>The value of <strong>number</strong> can be:</td>
</tr>
<tr>
<td></td>
<td>• 0—Unknown packets.</td>
</tr>
<tr>
<td></td>
<td>• 1—Hello packets.</td>
</tr>
<tr>
<td></td>
<td>• 2—DD packets.</td>
</tr>
<tr>
<td></td>
<td>• 3—Link state request packets.</td>
</tr>
<tr>
<td></td>
<td>• 4—Link state update packets.</td>
</tr>
<tr>
<td></td>
<td>• 5—Link state acknowledgement packets.</td>
</tr>
<tr>
<td></td>
<td>The value of <strong>string</strong> can be:</td>
</tr>
<tr>
<td></td>
<td>• Unknown.</td>
</tr>
<tr>
<td></td>
<td>• Hello.</td>
</tr>
<tr>
<td></td>
<td>• Database Description.</td>
</tr>
<tr>
<td></td>
<td>• Link State Request.</td>
</tr>
<tr>
<td></td>
<td>• Link State Update.</td>
</tr>
<tr>
<td></td>
<td>• Link State Acknowledgment.</td>
</tr>
<tr>
<td><strong>Router ID RouterID</strong></td>
<td>ID of the sending/receiving router.</td>
</tr>
<tr>
<td><strong>Area ID AreaID</strong></td>
<td>Area ID of the packet.</td>
</tr>
<tr>
<td><strong>Options number</strong></td>
<td>Router processing capabilities.</td>
</tr>
<tr>
<td><strong>RECV[DD]: LSA received Type(number), ID(LSID) is not recent</strong></td>
<td>Received LSA is not the latest.</td>
</tr>
<tr>
<td></td>
<td>number indicates the LSA number. It can be:</td>
</tr>
<tr>
<td></td>
<td>• 0x2001—Router-LSAs.</td>
</tr>
<tr>
<td></td>
<td>• 0x2002—Network-LSAs.</td>
</tr>
<tr>
<td></td>
<td>• 0x2003—Inter-Area-Prefix-LSAs.</td>
</tr>
<tr>
<td></td>
<td>• 0x2004—Inter-Area-Router-LSAs.</td>
</tr>
<tr>
<td></td>
<td>• 0x2005—AS-external-LSAs.</td>
</tr>
<tr>
<td></td>
<td>• 0x0008—Link-LSAs.</td>
</tr>
<tr>
<td></td>
<td>• 0x2009—Intra-Area-Prefix-LSAs.</td>
</tr>
<tr>
<td></td>
<td>• 0x000b—Grace-LSA.</td>
</tr>
<tr>
<td></td>
<td><strong>LSID</strong> indicates the Link-state ID of the LSA.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| RECV[DD]: Neighbor state is state, packet discarded | OSPFv3 discarded the received DD packet. state indicates the current neighbor state. It can be:  
  • Down—The neighbor is down.  
  • Attempt—A further attempt is made to contact the neighbor after receiving nothing from it.  
  • Init—The hello packet sent by the neighbor is received but 2-way communication is not established. |
| RECV[LS-Req]: Neighbor state is state packet discarded | The neighbor state is less than Exchange and the packet is discarded. state indicates the neighbor state. It can be:  
  • Down—The neighbor is down.  
  • Init—The hello packet sent by the neighbor is received but 2-way communication is not established.  
  • 2-Way—2-way communication is established.  
  • ExStart—The master/subordinate negotiation is performed. |
| [RECV]: Discard interface interface-type interface-number packet due to IPsec policy mismatch. IF SPI: if-spi, Packet SPI: packet-spi | OSPFv3 discarded the received packet because the packet does not match the IPsec policy on the receiving interface.  
  • if-spi—SPI on the interface.  
  • packet-spi—SPI carried in the packet. |

**Examples**

# Enable DD packet debugging on Router A. The output in this example was created when the following conditions exist:

- Router A is connected to Router B through Ethernet 1/0.
- OSPFv3 process 1 is created on Router A. OSPFv3 is enabled on Ethernet 1/0.
- OSPFv3 process 1 is created on Router B. OSPFv3 is enabled on Ethernet 1/0.

```bash
<Sysname> debugging ospfv3 packet dd
*0.31611969 Sysname RM/6/RMDEBUG:OSPFv3 Packet[SEND]:
src:(FE80::200:5EFF:FE01:B400) -> dst:(FE80::200:5EFF:FE01:B403)
// OSPFv3 sent DD packets from FE80::200:5EFF:FE01:B400 to FE80::200:5EFF:FE01:B403.
*0.290484 Sysname RM/6/RMDEBUG:OSPFv3 RECV[DD]: Negotiation done (Slave)
// OSPFv3 finished the master/subordinate negotiation. The current router works as the subordinate router.
```

**debugging ospfv3 route**

Use `debugging ospfv3 route` to enable OSPFv3 route debugging.

Use `undo debugging ospfv3 route` to disable OSPFv3 route debugging.
Syntax

depugging ospfv3 route [ ase | ia | install | spf ]
undo debugging ospfv3 route [ ase | ia | install | spf ]

Default

OSPFv3 route debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

ase: Specifies OSPFv3 ASE route debugging.
ia: Specifies OSPFv3 inter-area route debugging.
install: Specifies debugging of OSPFv3 routes installation into the routing table.
spf: Specifies OSPFv3 SPF route calculation debugging.

Usage guidelines

Table 9 describes output fields and messages for the debugging ospfv3 route command.

Table 120 Output from the debugging ospfv3 route command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPF/AreaID</td>
<td>SPF calculation.</td>
</tr>
<tr>
<td>string</td>
<td>indicates the LSDB containing the LSA. It can be:</td>
</tr>
<tr>
<td>Link, identified by interface type and interface number.</td>
<td></td>
</tr>
<tr>
<td>Area ID.</td>
<td></td>
</tr>
<tr>
<td>AS, indicating the flooding range is the entire AS.</td>
<td></td>
</tr>
<tr>
<td>number</td>
<td>indicates LSA number. It can be:</td>
</tr>
<tr>
<td>0x2001 — Router-LSAs.</td>
<td></td>
</tr>
<tr>
<td>0x2002 — Network-LSAs.</td>
<td></td>
</tr>
<tr>
<td>0x2003 — Inter-Area-Prefix-LSAs.</td>
<td></td>
</tr>
<tr>
<td>0x2004 — Inter-Area-Router-LSAs.</td>
<td></td>
</tr>
<tr>
<td>0x2005 — AS-external-LSAs.</td>
<td></td>
</tr>
<tr>
<td>0x0008 — Link-LSAs.</td>
<td></td>
</tr>
<tr>
<td>0x2009 — Intra-Area-Prefix-LSAs.</td>
<td></td>
</tr>
<tr>
<td>0x000b — Grace-LSA.</td>
<td></td>
</tr>
<tr>
<td>LSID</td>
<td>indicates link-state ID of the LSA.</td>
</tr>
<tr>
<td>adv_router</td>
<td>indicates ID of the advertising router.</td>
</tr>
<tr>
<td>* indicates that the LSA is self-originated.</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Install LS_TYPE Vertex Id</td>
<td>OSPFv3 installed the vertex to the candidate list.</td>
</tr>
<tr>
<td>LSID Adv by AdvRouter</td>
<td>LS_TYPE indicates LSA type. It can be:</td>
</tr>
<tr>
<td></td>
<td>• Router.</td>
</tr>
<tr>
<td></td>
<td>• Network.</td>
</tr>
<tr>
<td>candidate list</td>
<td>LSID indicates link State ID of the LSA.</td>
</tr>
<tr>
<td></td>
<td>AdvRouter indicates ID of the advertising router.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable SPF route debugging on one of the two interconnected devices enabled with OSPFv3. When you change the network topology, output similar to the following example is generated:

```
<Sysname> debugging ospfv3 route spf
*0.82470953 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Calculation timer scheduled [delay 5 secs]
// The calculation started 5 seconds later.
*0.82470953 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: SPF calculation timer is already scheduled
// The SPF calculation timer started.
*0.82475046 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Intra-Area SPF calculation timer expire
// The SPF calculation timer of area 0 expired.
*0.82475046 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Intra-Area SPF calculation (1st STAGE)
// SPF calculation in area 0 will be performed (first stage).
*0.82475046 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Get Router VertexId 0.0.0.0 Adv by 45.2.2.2
// OSPFv3 obtained the neighbor router vertex of the vertex whose router ID was 45.2.2.2 and whose Link State ID was 0.0.0.0.
*0.82475046 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Link[0] to Transit-Net 34.1.1.1 by interface 0.15.0.24:
// Link 0 was connected to neighbor 34.1.1.1 whose Link State ID was 0.15.0.24 over a transit network.
*0.82475046 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Install Network Vertex Id 0.15.0.24 Adv by 34.1.1.1 into candidate list
// The vertex with router ID 34.1.1.1 and Link State ID 0.15.0.24 was installed into the candidate list.
*0.82475062 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: No candidate node in the candidate list.
// There was no candidate node in the candidate list.
*0.82475062 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Calculate nexthop for (34.1.1.1-0.15.0.24)
// OSPFv3 was calculating the nexthop to the neighbor with router ID 34.1.1.1 and Link State ID 0.15.0.24.
*0.82475062 Sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Get Network Vertex Id 0.15.0.24 Adv by 34.1.1.1
// The vertex with Router ID 34.1.1.1 and Link State ID 0.15.0.24 got the next network vertex.
```
sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Link[0] to Router 34.1.1.1
// Link 0 pointed to router 34.1.1.1.

sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Link[0] (34.1.1.1-0.0.0.0):
doesn't have link back to (34.1.1.1-0.15.0.24)
// Vertex (34.1.1.1-0.0.0.0) did not have any link back to vertex 34.1.1.1-0.15.0.24.

sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]:
LSA[Area(0.0.0.0):Type(0x2001):0.0.0.0:45.2.2.2 *] is already in SPF tree
// The LSA (area ID is 0, type is 0x2001, router ID is 45.2.2.2, and Link State ID is 0.0.0.0) was already in the SPF tree.

sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Intra-Area SPF calculation (2nd STAGE)
// OSPFv3 performed SPF calculation (second stage).

sysname RM/6/RMDEBUG:OSPFv3 SPF[0.0.0.0]: Intra-Area SPF calculation (END)
// OSPFv3 finished intra-area SPF calculation.
PIM debugging commands

Support for VPN instances depends on your device model.

The output description tables in this document only contain fields and messages that require an explanation.

debugging pim

Use **debugging pim** to enable PIM debugging.

Use **undo debugging pim** to disable PIM debugging.

**Syntax**

```

```

**Default**

PIM debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **all-instance**: Specifies all MPLS L3VPN instances.
- **vpn-instance vpn-instance-name**: Specifies an MPLS L3VPN instance. The VPN instance name must be a case-sensitive string of 1 to 31 characters and must not contain any spaces.
- **all**: Specifies all types of PIM debugging.
- **assert**: Specifies PIM assert message debugging.
- **advanced-acl-number**: Specifies an advanced ACL number in the range of 3000 to 3999. An advanced ACL defines the range of multicast groups to which the advertised RP is designated.
- **receive**: Specifies inbound PIM packet debugging.
- **send**: Specifies outbound PIM packet debugging.
- **df**: Specifies DF information debugging.
- **event**: Specifies PIM event debugging.
- **join-prune**: Specifies PIM join/prune message debugging.
msdp: Specifies debugging for packets exchanged between PIM and MSDP.
neighbor: Specifies PIM neighbor debugging.
basic-acl-number: Specifies a basic ACL number in the range of 2000 to 2999.
register: Specifies PIM register message debugging.
routing-table: Specifies debugging for state change of the PIM multicast routing table.
rp: Specifies debugging for PIM packets related to the BSR and the RP.
state-refresh: Specifies PIM state refresh message debugging.

Usage guidelines

If neither all-instance nor vpn-instance is specified, this command enables PIM debugging for the public network.

Table 1 describes output fields and messages for the debugging pim assert command.

Table 121 Output from the debugging pim assert command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pref</td>
<td>Preference value.</td>
</tr>
<tr>
<td>metric</td>
<td>Metric value.</td>
</tr>
<tr>
<td>rpt set/unset</td>
<td>The RPT bit is 1/0.</td>
</tr>
<tr>
<td>reserved field non-zero</td>
<td>The reserved field is non-zero.</td>
</tr>
<tr>
<td>unknown neighbor</td>
<td>The neighbor is unknown.</td>
</tr>
<tr>
<td>truncated assert packet</td>
<td>The packet length is invalid.</td>
</tr>
<tr>
<td>bad group address/mask</td>
<td>The group address or the address mask is incorrect.</td>
</tr>
<tr>
<td>unknown group family</td>
<td>The group family is incorrect.</td>
</tr>
<tr>
<td>bad source address</td>
<td>The source address is incorrect.</td>
</tr>
<tr>
<td>locally scoped</td>
<td>Node-local or link-local scope.</td>
</tr>
<tr>
<td>Fsm:assert</td>
<td>Assert state machine.</td>
</tr>
<tr>
<td>current state</td>
<td>Current state of the assert state machine.</td>
</tr>
<tr>
<td>received event</td>
<td>Type of the event received by the assert state machine.</td>
</tr>
<tr>
<td>loser/Winner/noinfo</td>
<td>The assert state machine was in loser/winner/no-information state.</td>
</tr>
<tr>
<td>state1-&gt;state2</td>
<td>The assert state machine was transferred from state1 to state2.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the debugging pim df command.

Table 122 Output from the debugging pim df command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF election</td>
<td>DF election started.</td>
</tr>
<tr>
<td>DFT</td>
<td>DF election timer.</td>
</tr>
<tr>
<td>DFT expire time</td>
<td>Expiration time of DF election timer.</td>
</tr>
</tbody>
</table>
### Table 3: Output from the debugging pim event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unsupported pim version</td>
<td>The PIM packet version is not supported.</td>
</tr>
<tr>
<td>PIM packet too short</td>
<td>The PIM packet is too short.</td>
</tr>
<tr>
<td>checksum error</td>
<td>The checksum of the packet is incorrect.</td>
</tr>
<tr>
<td>non-pim interface</td>
<td>A PIM message was received on a non-PIM-enabled interface.</td>
</tr>
<tr>
<td>unsupported type</td>
<td>The specified PIM packet type is not supported.</td>
</tr>
<tr>
<td>Socket set option error</td>
<td>Socket option setting failed.</td>
</tr>
<tr>
<td>Packet send error</td>
<td>A PIM message failed to be sent out.</td>
</tr>
<tr>
<td>Source address is one of the interfaces address</td>
<td>The source address is the address of a local interface.</td>
</tr>
<tr>
<td>Source address address is invalid</td>
<td>The source address address is invalid.</td>
</tr>
<tr>
<td>Invalid source mask</td>
<td>The source address mask is invalid.</td>
</tr>
<tr>
<td>Active/Inactive event received</td>
<td>A source-active/inactive event was received.</td>
</tr>
<tr>
<td>Clear event received</td>
<td>A clear-entry event was received.</td>
</tr>
<tr>
<td>Wrong IIF</td>
<td>Incorrect incoming interface.</td>
</tr>
<tr>
<td>NoInfo</td>
<td>The downstream state machine was in no-information state.</td>
</tr>
<tr>
<td>PPending</td>
<td>The downstream state machine was in prune pending state.</td>
</tr>
<tr>
<td>Pruned</td>
<td>The PIM-DM downstream state machine was in pruned pending state.</td>
</tr>
<tr>
<td>Joined</td>
<td>The PIM-SM downstream state machine was in joined state.</td>
</tr>
<tr>
<td>Forwarding</td>
<td>The PIM-DM upstream state machine was in forwarded state.</td>
</tr>
<tr>
<td>Pruned</td>
<td>The PIM-DM upstream state machine was in pruned state.</td>
</tr>
<tr>
<td>AckPending</td>
<td>The PIM-DM upstream state machine was in ack-pending state.</td>
</tr>
<tr>
<td>Joined</td>
<td>The PIM-SM (S, G) or (*, G) upstream state machine was in joined state.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NotJoined</td>
<td>The PIM-SM (S, G) or (*, G) upstream state machine was in not-joined state.</td>
</tr>
<tr>
<td>PruneTmp</td>
<td>The PIM-SM (S, G, RPT) downstream state machine was in temporary prune state.</td>
</tr>
<tr>
<td>PPendingTmp</td>
<td>The PIM-SM (S, G, RPT) downstream state machine was in temporary prune pending state.</td>
</tr>
<tr>
<td>PPT Expired</td>
<td>The prune pending timer timed out.</td>
</tr>
<tr>
<td>RPF_Interface changed</td>
<td>The RPF interface changed.</td>
</tr>
<tr>
<td>Genid changed</td>
<td>The neighbor generation ID changed.</td>
</tr>
<tr>
<td>PT Expired</td>
<td>The prune timer timed out.</td>
</tr>
<tr>
<td>Failed to pass MSF</td>
<td>Failed to pass multicast source filtering.</td>
</tr>
<tr>
<td>NotOriginator/Originator</td>
<td>The originator state machine was in not-originator/originator state.</td>
</tr>
<tr>
<td>SAT Expired</td>
<td>The source-alive timer timed out.</td>
</tr>
<tr>
<td>Join suppressed</td>
<td>The device received a join message to the upstream neighbor on the incoming interface and suppressed its own join message.</td>
</tr>
<tr>
<td>Override it</td>
<td>The device received a prune message to the upstream neighbor on the incoming interface and sent a join message.</td>
</tr>
<tr>
<td>ET Expired</td>
<td>The PIM-SM downstream interface timer timed out.</td>
</tr>
<tr>
<td>register downstream</td>
<td>Registering the outgoing interface.</td>
</tr>
<tr>
<td>Mcast-Boundary-Changed</td>
<td>Multicast boundary change event.</td>
</tr>
<tr>
<td>flush timer</td>
<td>Timer used for periodically checking whether new RP and DF information issued to the forwarding module exist.</td>
</tr>
<tr>
<td>walk timer</td>
<td>Timer used for periodically traversing RP table entries.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the **debugging pim join-prune** command.

**Table 124 Output from the debugging pim join-prune command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP</td>
<td>Join/prune message.</td>
</tr>
<tr>
<td>GFT</td>
<td>Graft message.</td>
</tr>
<tr>
<td>GAK</td>
<td>Graft-ack message.</td>
</tr>
<tr>
<td>bad group/source address, mask or family</td>
<td>Incorrect group/source address, mask or family.</td>
</tr>
<tr>
<td>Upstream</td>
<td>Upstream neighbor information in the message.</td>
</tr>
<tr>
<td>Groups</td>
<td>Number of groups in the message.</td>
</tr>
<tr>
<td>Group: addr/mask -- m joins n prunes</td>
<td>Group information in the message, including group address/mask length, m times of joins, and n times of prunes.</td>
</tr>
<tr>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Join: addr/mask flag</td>
<td></td>
</tr>
<tr>
<td>Join: source address/mask flag.</td>
<td></td>
</tr>
<tr>
<td>Message truncated. Ignoring message</td>
<td></td>
</tr>
<tr>
<td>The message was dropped due to invalid message length.</td>
<td></td>
</tr>
<tr>
<td>Unable to decode address</td>
<td></td>
</tr>
<tr>
<td>Address decoding failed.</td>
<td></td>
</tr>
<tr>
<td>Upstream neighbor is not this router. Ignoring</td>
<td></td>
</tr>
<tr>
<td>The message was dropped because the upstream neighbor is not this device.</td>
<td></td>
</tr>
<tr>
<td>group boundary detected for address1 on address2</td>
<td></td>
</tr>
<tr>
<td>Address 1 is within the multicast boundary configured on the interface corresponding to address 2.</td>
<td></td>
</tr>
<tr>
<td>Group address1 ignored in message on address2</td>
<td></td>
</tr>
<tr>
<td>Address 1 is within the multicast boundary configured on the interface corresponding to address 2. This group is ignored.</td>
<td></td>
</tr>
<tr>
<td>Message from unknown neighbor</td>
<td></td>
</tr>
<tr>
<td>A message was received from an unknown neighbor.</td>
<td></td>
</tr>
<tr>
<td>Join/Prune received for non-local neighbor</td>
<td></td>
</tr>
<tr>
<td>A join/prune message for a non-local upstream neighbor was received.</td>
<td></td>
</tr>
<tr>
<td>Override timer expires</td>
<td></td>
</tr>
<tr>
<td>The prune override timer timed out.</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the **debugging pim neighbor** command.

**Table 125 Output from the debugging pim neighbor command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEL</td>
<td>PIM hello message.</td>
</tr>
<tr>
<td>hello packet</td>
<td>PIM hello message.</td>
</tr>
<tr>
<td>Option: (m), length: (n)</td>
<td>PIM hello message option: option value (specified by (m)) and option length: length value (specified by (n)).</td>
</tr>
<tr>
<td>Holdtime:</td>
<td>Holdtime field of the PIM hello message.</td>
</tr>
<tr>
<td>Tbit</td>
<td>Tbit option.</td>
</tr>
<tr>
<td>Lan delay</td>
<td>LAN delay option.</td>
</tr>
<tr>
<td>Override interval</td>
<td>Override interval option.</td>
</tr>
<tr>
<td>DR priority</td>
<td>DR priority option.</td>
</tr>
<tr>
<td>Genid</td>
<td>Generation ID option.</td>
</tr>
<tr>
<td>Version</td>
<td>Version field of the state refresh option.</td>
</tr>
<tr>
<td>Refresh interval</td>
<td>State refresh interval field of the state refresh option.</td>
</tr>
<tr>
<td>Reserved</td>
<td>Reserved field of the state refresh option.</td>
</tr>
<tr>
<td>Secondary address(es)</td>
<td>Address in the address list option.</td>
</tr>
<tr>
<td>Unknown option value</td>
<td>The option value is unknown.</td>
</tr>
<tr>
<td>without SR capability</td>
<td>No state refresh capability.</td>
</tr>
<tr>
<td>Elected as DR on interface interfacename</td>
<td>The device was elected as the DR in the network attached to <em>interfacename</em>.</td>
</tr>
<tr>
<td>Unelected as DR on interface interfacename</td>
<td>The device was no longer the DR for the network attached to <em>interfacename</em>.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
**PIM Neighbor address on interface interfacename timed out** | Neighbor address on interfacename timed out.

Failed to create PIM neighbor address, PIM Hello denoted by neighbor policy on interfacename | A PIM neighbor relationship failed to be established because the source address in the hello message failed to pass the configured hello message filtering.

Table 6 describes output fields and messages for the **debugging pim register** command.

**Table 126 Output from the debugging pim register command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG</td>
<td>Register message.</td>
</tr>
<tr>
<td>RSP</td>
<td>Register-stop message.</td>
</tr>
<tr>
<td>Register Stop</td>
<td>Register stopped.</td>
</tr>
<tr>
<td>Border bit</td>
<td>Boundary bit.</td>
</tr>
<tr>
<td>src</td>
<td>Source address of the IP packet.</td>
</tr>
<tr>
<td>dst</td>
<td>Destination address of the IP packet.</td>
</tr>
<tr>
<td>probe</td>
<td>Probe message.</td>
</tr>
<tr>
<td>ignored</td>
<td>A message was dropped.</td>
</tr>
</tbody>
</table>

Table 7 describes output fields and messages for the **debugging pim routing-table** command.

**Table 127 Output from the debugging pim routing-table command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating/Deleting</td>
<td>PIM created or deleted entries.</td>
</tr>
<tr>
<td>mrt</td>
<td>Multicast routing table.</td>
</tr>
<tr>
<td>Add/Del oil</td>
<td>PIM added or deleted outgoing interfaces.</td>
</tr>
<tr>
<td>Null iif</td>
<td>Null incoming interface.</td>
</tr>
<tr>
<td>Adding/Deleting iif</td>
<td>PIM added or deleted incoming interfaces.</td>
</tr>
<tr>
<td>RP is not found</td>
<td>The RP was not found.</td>
</tr>
</tbody>
</table>

Table 8 describes output fields and messages for the **debugging pim rp** command.

**Table 128 Output from the debugging pim rp command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto-RP announce</td>
<td>Auto-RP announce message.</td>
</tr>
<tr>
<td>auto-RP discovery</td>
<td>Auto-RP discovery message.</td>
</tr>
<tr>
<td>C-RP/CRP</td>
<td>Candidate RP.</td>
</tr>
<tr>
<td>BSR</td>
<td>Bootstrap router.</td>
</tr>
<tr>
<td>prefix count</td>
<td>Prefix count field in the C-RP advertisement message.</td>
</tr>
<tr>
<td>priority</td>
<td>Priority field in the C-RP advertisement message.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>holdtime</td>
<td>Holdtime field in the C-RP advertisement message.</td>
</tr>
<tr>
<td>Admin Scope Zone</td>
<td>Administratively scoped zone.</td>
</tr>
<tr>
<td>Bad BSR address</td>
<td>Incorrect BSR address.</td>
</tr>
<tr>
<td>frag</td>
<td>Fragment tag field in the BSR bootstrap message.</td>
</tr>
<tr>
<td>pri</td>
<td>Priority field in the BSR bootstrap message.</td>
</tr>
<tr>
<td>hash mask len</td>
<td>Hash mask length field in the BSR bootstrap message.</td>
</tr>
<tr>
<td>Group address/length: frags m, C-RP’s n</td>
<td>The frags field corresponding to address/length in the BSR bootstrap message is specified by m. The number of C-RPs is specified by n.</td>
</tr>
<tr>
<td>address pri: m, holdtime: n</td>
<td>The priority of C-RP address in the BSR bootstrap message is specified by m and holdtime is specified by n.</td>
</tr>
</tbody>
</table>
| Auto-RP discovery packet: RP agent address, RP count m, Holdtime n | An auto-RP discovery message was received. Information about the auto-RP discovery message includes the following:  
  • RP agent specified by address.  
  • RP count specified by m.  
  • Holdtime specified by n. |
<p>| delete RP-Set                    | An RP set was deleted.                                                      |
| too short length                 | Message length is too short.                                                |
| bad group address                | Incorrect group address.                                                    |
| bad group mask length            | Incorrect group mask length.                                                |
| bad BSR address                  | Incorrect BSR address.                                                      |
| bad BSR address family           | Incorrect BSR address family.                                               |
| bad BSR hash mask length         | Incorrect BSR hash mask length.                                             |
| bad scope zone mask              | Incorrect scope zone mask.                                                 |
| Unknown group address family     | Incorrect group address family.                                             |
| not directly connected source    | The source is not directly connected.                                       |
| unknown neighbor                 | The neighbor is unknown.                                                    |
| Bad frag-rp-count field          | Incorrect frag-rp-count field in the BSR bootstrap message.                |
| Bad frag-rp field length         | Incorrect total length of frag-rp fields in the BSR bootstrap message.     |
| BSR mechanism                    | BSR mechanism independent of administrative scoping.                       |
| Upstream to BSR                  | Upstream toward the BSR.                                                    |
| no BSR is available              | No BSR is available.                                                       |
| add/remove register vif          | A register virtual interface was added or removed.                          |
| Expiring CRP                     | C-RP that aged out.                                                        |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lose the ASBSR election</td>
<td>The device lost the BSR election in the BSR admin-scope region.</td>
</tr>
<tr>
<td>Lose the BSR election</td>
<td>The device lost the BSR election.</td>
</tr>
<tr>
<td>locally scoped</td>
<td>Node-local or link-local scope.</td>
</tr>
<tr>
<td>RP changed</td>
<td>The RP changed.</td>
</tr>
<tr>
<td>pending state</td>
<td>The BSR changed to the pending state.</td>
</tr>
<tr>
<td>Update the BSR’s state to elected</td>
<td>The BSR changed to the elected state.</td>
</tr>
<tr>
<td>RPF Failure</td>
<td>RPF check failed.</td>
</tr>
<tr>
<td>admin scope multicast address</td>
<td>Address in the admin-scope range.</td>
</tr>
</tbody>
</table>

Table 9 describes output fields and messages for the `debugging pim state-refresh` command.

Table 129 Output from the debugging pim state-refresh command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRM</td>
<td>State refresh message.</td>
</tr>
<tr>
<td>Message truncated</td>
<td>Message length is invalid.</td>
</tr>
<tr>
<td>bad group address</td>
<td>Incorrect group address.</td>
</tr>
<tr>
<td>invalid group mask length</td>
<td>Incorrect group mask length.</td>
</tr>
<tr>
<td>Originator address</td>
<td>Address of the state refresh message originator.</td>
</tr>
<tr>
<td>preference</td>
<td>Preference field of the message.</td>
</tr>
<tr>
<td>metric</td>
<td>Metric field of the message.</td>
</tr>
<tr>
<td>mask length</td>
<td>Mask length field of the message.</td>
</tr>
<tr>
<td>ttl</td>
<td>TTL value of the message.</td>
</tr>
<tr>
<td>prune indicator</td>
<td>Prune indicator flag bit.</td>
</tr>
<tr>
<td>prune now</td>
<td>Prune now indicator flag bit.</td>
</tr>
<tr>
<td>assert override</td>
<td>Assert override flag bit.</td>
</tr>
</tbody>
</table>

Examples

The output in the following examples was created under the condition that PIM-DM or PIM-SM is enabled:

# Enable debugging for inbound PIM assert messages for the public network.

```
<Sysname> debugging pim assert receive
```

```
*0.594609 Sysname PIM/7/ASSERT:(public net): PIM ver 2 AST receiving 30.1.1.2 ->224.0.0.13 on Vlan-interface30 (P012343)
```

```
*0.594609 Sysname PIM/7/ASSERT:(public net): For 229.0.0.1/32 from 100.1.1.11, rpt unset, pref 10, metric 3 (P012351)
```

```
// PIM received a PIMv2 assert message from 30.1.1.2 to 224.0.0.13 on VLAN-interface 30.
```

# Enable debugging for outbound PIM assert messages for the public network.

```
<Sysname> debugging pim assert send
```

203
// PIM sent a PIMv2 assert message from 30.1.1.2 to 224.0.0.13 out of VLAN-interface 30.

# Enable DF information debugging on the public network.
<Sysname> debugging pim df

// The RP for BIDIR-PIM is 1.1.1.1. A DF election was triggered on VLAN-interface 2. The DF election timer was started. After the timer expired, an Offer packet was sent and MC was set to 1.

// PIM received a better winner packet on VLAN-interface 2. The DF state changed from Offer to Lose. The DF was set to 2.1.1.1, the DF election timer was stopped, and the DF of the RP was issued to the driver.

# Enable debugging for PIM events for the public network.
// Because the prune pending timer timed out, the downstream state machine of the outgoing interface VLAN-interface 10 of the PIM-SM state entry (*, 229.0.0.1) transited from prune pending to no-information.

# Enable debugging for join/prune messages for the public network.
<Sysname> debugging pim join-prune receive
*0.6209953 Sysname PIM/7/JP:(public net): PIM ver 2 JP receiving 20.1.1.2 -> 224.0.0.13 on Vlan-interface20 (P012176)
*0.6209953 Sysname PIM/7/JP:(public net): Upstream 20.1.1.1, Groups 1, Holdtime 210 (P012178)
*0.6209953 Sysname PIM/7/JP:(public net): Group: 229.0.0.1/32 --- 1 joins 0 prunes (P012184)
*0.6209953 Sysname PIM/7/JP:(public net): Join: 100.1.1.11/32 S (P012188)

// PIM received a join/prune message from 20.1.1.2 to 224.0.0.13 on VLAN-interface 20.

# Enable debugging for outbound PIM join/prune messages for the public network.
<Sysname> debugging pim join-prune send
*0.5916437 Sysname PIM/7/JP:(public net): PIM ver 2 JP sending 20.1.1.2 -> 224.0.0.13 on Vlan-interface20 (P012176)
*0.5916437 Sysname PIM/7/JP:(public net): Upstream 20.1.1.1, Groups 1, Holdtime 210 (P012178)
*0.5916437 Sysname PIM/7/JP:(public net): Group: 229.0.0.1/32 --- 1 joins 0 prunes (P012184)
*0.5916437 Sysname PIM/7/JP:(public net): Join: 100.1.1.11/32 S (P012188)

// PIM sent a join/prune message from 20.1.1.2 to 224.0.0.13 out of VLAN-interface 2.

<Sysname> debugging pim neighbor receive
*0.2605047 Sysname PIM/7/NBR:(public net): PIM ver 2 HEL receiving 10.1.1.1 -> 224.0.0.13 on Vlan-interface10 (P011857)
*0.2605047 Sysname PIM/7/NBR:(public net): Option: 1, length: 2 (P011891)
*0.2605062 Sysname PIM/7/NBR:(public net): Holdtime: 105 (P011898)
*0.2605062 Sysname PIM/7/NBR:(public net): Option: 2, length: 4 (P011891)
*0.2605062 Sysname PIM/7/NBR:(public net): Tbit: unset (P011907)
*0.2605062 Sysname PIM/7/NBR:(public net): Lan delay: 500 (P011908)
*0.2605062 Sysname PIM/7/NBR:(public net): Override interval: 2500 (P011909)
*0.2605062 Sysname PIM/7/NBR:(public net): Option: 19, length: 4 (P011891)
*0.2605062 Sysname PIM/7/NBR:(public net): DR priority: 1 (P011916)
*0.2605094 Sysname PIM/7/NBR:(public net): Genid: 0x67dc7fbb (P011923)

// PIM received a PIMv2 hello message from 10.1.1.1 to 224.0.0.13 on VLAN-interface 10.

# Enable debugging for outbound packets related to PIM neighbors for the public network.
<Sysname> debugging pim neighbor send
*0.1526312 Sysname PIM/7/NBR:(public net): PIM ver 2 HEL sending 10.1.1.1 -> 224.0.0.13 on Vlan-interface10 (P011857)
*0.1526312 Sysname PIM/7/NBR:(public net): Option: 1, length: 2 (P011891)
*0.1526312 Sysname PIM/7/NBR:(public net): Holdtime: 105 (P011898)
*0.1526328 Sysname PIM/7/NBR:(public net): Option: 2, length: 4 (P011891)
*0.1526328 Sysname PIM/7/NBR:(public net): Tbit: unset (P011907)
*0.1526343 Sysname PIM/7/NBR:(public net): Lan delay: 500 (P011908)
*0.1526343 Sysname PIM/7/NBR:(public net): Override interval: 2500 (P011909)
// PIM sent a PIMv2 hello message from 10.1.1.1 to 224.0.0.13 out of VLAN-interface 10.

# Enable debugging for inbound PIM register messages for the public network.
<Sysname> debugging pim register receive
*0.4825671 Sysname PIM/7/REG:(public net): PIM ver 2 RSP receiving 12.12.12.12 -> 100.1.1.1 on Vlan-interface20 (S01648)
*0.4825671 Sysname PIM/7/REG:(public net): Register Stop: (100.1.1.10, 229.0.0.1) (S01650)

// PIM received a register-stop message from 12.12.12.12 to 100.1.1.1 on VLAN-interface 20. In the
register-stop message, the group address is 229.0.0.1 and the multicast source address is 100.1.1.10.

# Enable debugging for outbound PIM register messages for the public network.
<Sysname> debugging pim register send
*0.5554875 Sysname PIM/7/REG:(public net): PIM ver 2 REG sending 100.1.1.1 -> 12.12.12.12 on Vlan-interface20 (S01729)
*0.5554875 Sysname PIM/7/REG:(public net): Border bit: false Null bit: true (S01737)
*0.5554875 Sysname PIM/7/REG:(public net): Encapsulated ip src: 100.1.1.11, dst:229.0.0.1, len: 20 (S01744)

// PIM sent a PIMv2 register message from 100.1.1.1 to 12.12.12.12 out of VLAN-interface 20. The border
bit is not set, but Null bit is set.

# Enable debugging for PIM multicast routing table state changes for the public network.
<Sysname> debugging pim routing-table
*0.4825125 Sysname PIM/7/ROUT:(public net): PIM-SM: Create (100.1.1.10, 229.0.0.1) entry in mrt. (S018118)
*0.4825125 Sysname PIM/7/ROUT:(public net): PIM-SM: Adding iif 100.1.1.1 to (100.1.1.10, 229.0.0.1). (S012740)
*0.4825125 Sysname PIM/7/ROUT:(public net): PIM-SM: Add oif: register for (100.1.1.10, 229.0.0.1) (S017199)

// PIM created a PIM-SM state entry (100.1.1.10, 229.0.0.1), with the interface toward 100.1.1.1 as the
incoming interface and the registered virtual interface as the outgoing interface.

# Enable debugging for inbound PIM packets related to the BSR and the RP for the public network.
<Sysname> debugging pim rp receive
*0.2776421 Sysname PIM/7/RP:(public net): PIM ver 2 CRP receiving 10.1.1.2 -> 19.1.1.1 on Vlan-interface10 (S01409)
*0.2776421 Sysname PIM/7/RP:(public net): C-RP 12.12.12.12, prefix count 1, priority: 0, holdtime 150 (S01433)
*0.2776421 Sysname PIM/7/RP:(public net): 224.0.0.0/4 (S01454)
*0.2776421 Sysname PIM/7/RP:(public net): Admin Scope Zone: 0 (S01455)

// PIM received a PIMv2 C-RP advertisement message from 10.1.1.2 to 19.1.1.1 on VLAN-interface 10.
PIM ver 2 BSR receiving 10.1.1.1 -> 224.0.0.13 on Vlan-interface10 (S01500)
*0.3307890 Sysname PIM/7/RP:(public net): BSR 19.1.1.1, frag 0, pri 0, hash mask len 30 (S01520)
*0.3307890 Sysname PIM/7/RP:(public net): Group 224.0.0.0/4: frags 1, C-RP's 1 (S01544)
*0.3307890 Sysname PIM/7/RP:(public net): 12.12.12.12 pri: 0, holdtime: 150 (S01568)
Sysname PIM/7/RP:(public net): Receive BSR packet. And use BSR Mechanism now. (S031694)

// PIM received a PIMv2 bootstrap message from 10.1.1.1 to 224.0.0.13 on VLAN-interface 10.

# Enable debugging for outbound PIM packets related to BSR and RP for the public network.
<Sysname> debugging pim rp send

*0.3303375 Sysname PIM/7/RP:(public net): PIM ver 2 CRP sending 10.1.1.2 -> 19.1.1.1 on Vlan-interface10 (S01409)
*0.3303375 Sysname PIM/7/RP:(public net): C-RP 12.12.12.12, prefix count 1, priority: 0, holdtime 150 (S01433)
*0.3303375 Sysname PIM/7/RP:(public net): 224.0.0.0/4 (S01454)
*0.3303375 Sysname PIM/7/RP:(public net): Admin Scope Zone: 0 (S01455)

// PIM sent a PIMv2 C-RP advertisement message from 10.1.1.2 to 19.1.1.1 out of VLAN-interface 10.
*0.2780906 Sysname PIM/7/RP:(public net): PIM ver 2 BSR sending 20.1.1.1 -> 224.0.0.13 on Vlan-interface20 (S01500)
*0.2780906 Sysname PIM/7/RP:(public net): BSR 19.1.1.1, frag 0, pri 0, hash masklen 30 (S01520)
*0.2780906 Sysname PIM/7/RP:(public net): Group 224.0.0.0/4: frags 2, C-RP's 2 (S01544)
*0.2780906 Sysname PIM/7/RP:(public net): 10.1.1.2 pri: 0, holdtime: 150 (S01568)
*0.2780906 Sysname PIM/7/RP:(public net): 12.12.12.12 pri: 0, holdtime: 150 (S01568)

// PIM sent a PIMv2 BSR bootstrap message from 20.1.1.1 to 224.0.0.13 out of VLAN-interface 20.

# Enable debugging for inbound PIM state refresh packets for the public network.
<Sysname> debugging pim state-refresh receive

*0.8386062 Sysname PIM/7/SRM:(public net): PIM ver 2 SRM receiving 20.1.1.1 -> 224.0.0.13 on Vlan-interface20 (D19670)
*0.8386062 Sysname PIM/7/SRM:(public net): Group address: 229.0.0.1/32 flags: 00000000 (D19672)
*0.8386062 Sysname PIM/7/SRM:(public net): Source address: 100.1.1.11 (D19674)
*0.8386062 Sysname PIM/7/SRM:(public net): Originator address: 20.1.1.1 (D19676)
*0.8386062 Sysname PIM/7/SRM:(public net): preference: 0 metric: 0 mask length: 0 (D19696)
*0.8386062 Sysname PIM/7/SRM:(public net): ttl: 254 prune indicator: unset prunenow: set assert override: set (D19699)
*0.8386062 Sysname PIM/7/SRM:(public net): interval: 1 (D19700)

// PIM received a state refresh packet from 20.1.1.1 to 224.0.0.13 on VLAN-interface 20.

# Enable debugging for outbound PIM state refresh packets for the public network.
<Sysname> debugging pim state-refresh send

*0.8345296 Sysname PIM/7/SRM:(public net): PIM ver 2 SRM sending 20.1.1.1 -> 224.0.0.13 on Vlan-interface20 (D19670)
*0.8345296 Sysname PIM/7/SRM:(public net): Group address: 229.0.0.1/32 flags: 00000000 (D19672)
*0.8345296 Sysname PIM/7/SRM:(public net): Source address: 100.1.1.11 (D19674)
*0.8345296 Sysname PIM/7/SRM:(public net): Originator address: 20.1.1.1 (D19676)
*0.8345296 Sysname PIM/7/SRM:(public net): preference: 0 metric: 0 mask length: 0 (D19696)
*0.8345312 Sysname PIM/7/SRM:(public net): ttl: 254 prune indicator: unset prunenow: unset assert override: set (D19699)
*0.8345312 Sysname PIM/7/SRM:(public net): interval: 1 (D19700)

// PIM received a state refresh packet from 20.1.1.1 to 224.0.0.13 on VLAN-interface 2.
PKI debugging commands

debugging pki

Use debugging pki to enable PKI debugging.
Use undo debugging pki to disable PKI debugging.

Syntax

debugging pki { all | certificate access-control-policy | error | request | retrieval | verify }
undo debugging pki { all | certificate | error | request | retrieval | verify }

Default
PKI debugging is disabled.

Views
User view

Default command level
1: Monitor level

Parameters

all: Specifies all types of IPsec debugging.
certificate access-control-policy: Specifies debugging for certificate attribute-based access control policies.
error: Specifies PKI error debugging.
request: Specifies certificate request debugging.
retrieval: Specifies certificate retrieval debugging.
verify: Specifies certificate validation debugging.

Usage guidelines

Table 1 to Table 4 describe output fields and messages for the PKI debugging commands.

Table 130 Output from the debugging pki certificate access-control-policy command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>the attribute number Match in attribute group 'group-number'</td>
<td>The certificate matches the attribute number in certificate attribute group group-number.</td>
</tr>
<tr>
<td>the attribute number Not Match in attribute group 'group-number'</td>
<td>The certificate does not match the attribute number in certificate attribute group group-number.</td>
</tr>
<tr>
<td>Not match the rule number in access control policy 'string'. Check the next rule.</td>
<td>The certificate did not match rule number in access control policy string. Trying the next rule.</td>
</tr>
</tbody>
</table>
Table 131 Output from the debugging pki error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCEP receive message: wrong MIME content type</td>
<td>Type of the content of the SCEP packet received from the CA is not correct.</td>
</tr>
<tr>
<td>Error while sending message</td>
<td>Error occurred when SCEP packet is sent.</td>
</tr>
<tr>
<td>PKCS#7 develope: reason</td>
<td>Reason for PKCS#7 de-encapsulation failure.</td>
</tr>
<tr>
<td>PKCS#7 develope: illegal size of payload</td>
<td>The size of the payload in the de-encapsulated packet is illegal.</td>
</tr>
<tr>
<td>Certificate enroll failed, error code is 72</td>
<td>Certificate request failed. The error code is 72, which means that the key pair has been used before.</td>
</tr>
</tbody>
</table>

Table 132 Output from the debugging pki request command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKCS#7 envelope</td>
<td>PKCS#7 encapsulation.</td>
</tr>
<tr>
<td>PKCS#7 develope</td>
<td>PKCS#7 de-encapsulation.</td>
</tr>
<tr>
<td>SCEP send message</td>
<td>The PKI module sent a message to the CA through SCEP.</td>
</tr>
<tr>
<td>SCEP receive message</td>
<td>The PKI module received a message from the CA through SCEP.</td>
</tr>
<tr>
<td>PKI Get the Signed Certificates</td>
<td>The entity obtained the signed certificates.</td>
</tr>
<tr>
<td>issuer</td>
<td>DN of the certificate issuer.</td>
</tr>
</tbody>
</table>

Table 133 Output from the debugging pki retrieval command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI GetCert CA&amp;RA</td>
<td>The PKI module obtained a CA/RA certificate chain from the CA server.</td>
</tr>
<tr>
<td>PKCS#7 envelope</td>
<td>PKCS#7 encapsulation.</td>
</tr>
<tr>
<td>PKCS#7 develope</td>
<td>PKCS#7 de-encapsulation.</td>
</tr>
<tr>
<td>SCEP send message</td>
<td>The PKI module sent a message to the CA through SCEP.</td>
</tr>
<tr>
<td>SCEP receive message</td>
<td>The PKI module received a message from the CA through SCEP.</td>
</tr>
</tbody>
</table>

Examples

# Configure a certificate attribute-based access control policy and log in to the device through IE using HTTPS. The IE client certificate is as shown in Figure 1.
# Configure a certificate attribute group and a certificate attribute-based access control policy on the device.

```bash
<Sysname> system-view
<Sysname> pki certificate attribute-group 1
<Sysname-pki-cert-attribute-group-1> attribute 1 issuer-name dn ctn ssl
<Sysname-pki-cert-attribute-group-1> attribute 2 issuer-name dn nctn hhh01
<Sysname-pki-cert-attribute-group-1> attribute 3 subject-name dn ctn ssl-client
<Sysname> pki certificate access-control-policy 1
<Sysname-pki-cert-acp-1> rule 1 permit 1
<Sysname-pki-cert-acp-1> quit
```

# Enable HTTPS service.

```bash
<Sysname> ip https certificate access-control-policy 1
<Sysname> ip https-ssl-server-policy ssl
<Sysname> ip https enable
<Sysname> quit
```

# Enable debugging for certificate attribute-based access control policy and then log in through IE.

```
<Sysname> debugging pki certificate access-control-policy
*0.35979976 Sysname PKI/7/PKI_Debug:PKI_Certificate: the attribute 1 Match in attribute group '1'. Check the next attribute.
*0.35980120 Sysname PKI/7/PKI_Debug:PKI_Certificate: the attribute 2 Match in attribute group '1'. Check the next attribute.
```
// The client certificate passed the certificate validation after it matched Attributes 1, 2, and 3 in the certificate attribute group.

# Disable HTTPS service and add a new certificate attribute group.
<Sysname> system-view
[Sysname] undo ip https enable
[Sysname] pki certificate attribute-group 2
[Sysname-cert-attribute-group-2] attribute 1 issuer-name dn nctn ssl
[Sysname-cert-attribute-group-2] attribute 2 issuer-name dn nctn hhh01
[Sysname-cert-attribute-group-2] attribute 3 subject-name dn ctn ssl-client
[Sysname-cert-attribute-group-2] quit
[Sysname] pki certificate access-control-policy 2
[Sysname-cert-acp-2] rule 1 permit 2
[Sysname-cert-acp-2] quit

# Enable HTTPS service.
[Sysname] ip https certificate access-control-policy 2
[Sysname] ip https ssl-server-policy ssl
[Sysname] ip https enable

# Log in by using an IE browser.
*0.38231901 Sysname PKI/7/PKI_Debug:PKI_Certificate: the attribute 1 Not Match in attribute group '2'.
*0.38232030 Sysname PKI/7/PKI_Debug:PKI_Certificate: Not match the rule 1 in access control policy '2'. Check the next rule.
*0.38232190 Sysname PKI/7/PKI_Debug:PKI_Certificate: Certificate doesn't match any rules in access control policy '2'. Access Deny

// The client certificate failed the certificate validation after it failed to match Attribute 1, 2, or 3 of the certificate attribute group.

# Disable HTTPS service and apply multiple certificate attribute groups to the certificate attribute-based access control policy.
[Sysname] undo ip https enable
[Sysname] pki certificate access-control-policy 2
[Sysname-cert-acp-2] rule 1 permit 2
[Sysname-cert-acp-2] rule 2 permit 1
[Sysname-cert-acp-2] quit

# Enable HTTPS service.
[Sysname] ip https certificate access-control-policy 2
[Sysname] ip https ssl-server-policy ssl
[Sysname] ip https enable

# Log in by using an IE browser.
*0.38011098 Sysname PKI/7/PKI_Debug:PKI_Certificate: the attribute 1 Not Match in attribute group '2'.
*0.38011221 Sysname PKI/7/PKI_Debug:PKI_Certificate: Not match the rule 1 in access control policy '2'. Check the next rule.
*0.38011383 Sysname PKI/7/PKI_Debug:PKI_Certificate: the attribute 1 Match in
attribute group '1'. Check the next attribute.
*0.38011540 Sysname PKI/7/PKI_Debug:PKI_Certificate: the attribute 2 Match in
 attribute group '1'. Check the next attribute.
*0.38011706 Sysname PKI/7/PKI_Debug:PKI_Certificate: the attribute 3 Match in
 attribute group '1'. Check the next attribute.
*0.38011860 Sysname PKI/7/PKI_Debug:PKI_Certificate: Match the rule id: 2, ac
 tion: permit in access control policy '2'. Access Permit

// The client certificate passed the client certificate validation after failing to match Attribute 1 but
 successfully matching the second rule in the policy.

# Enable PKI certificate retrieval debugging. When a local certificate is requested after a CA certificate
 is retrieved, output similar to the following example is generated:

<Sysname> debugging pkix retrieval
[Sysname] pkix retrieval-certificate ca domain crt
Retrieving CA/RA certificates. Please wait a while........
*0.507125 Sysname PKI/7/PKI_Debug:Host: 4.4.4.133
// The host IP address of the CA server is 4.4.4.133.
*0.507141 Sysname PKI/7/PKI_Debug:Port: 446
// The port number of the SCEP protocol is 446.
*0.507141 Sysname PKI/7/PKI_Debug:Path: 6953bf7fb5b1cf514376243ce67ebed1209c292a
// The output shoes the path of the CA server.
*0.507157 Sysname PKI/7/PKI_Debug:HTTP request message is: GET
/6953bf7fb5b1cf514376243ce67ebed1209c292a/pkiclient.exe?operation=GetCACert&message=r
sa HTTP/1.0
// The content of the HTTP request message is "getting the CA certificate".
*0.507157 Sysname PKI/7/PKI_Debug:Start to send message............
*0.507157 Sysname PKI/7/PKI_Debug:SCEP send message:IP = 0x85040404
// The PKI module sent a packet to the CA through SCEP.
The trusted CA's finger print is:
        MD5 fingerprint:8FDC C669 7A95 5505 8C0A 8633 818D A0A1
        SHA1 fingerprint:8CCD 07AD 6C9A 229B 3378 2430 F038 A142 D175 190E
// The certificate fingerprint of the CA was calculated by using the hash algorithm.
Is the finger print correct?(Y/N):
*0.507250 Sysname PKI/7/PKI_Debug:SCEP receive message: Server returned status code 200
// The PKI module received a status code from the CA server through SCEP, which means OK.
*0.507266 Sysname PKI/7/PKI_Debug:Get CA certificates: received 1 certificates.
// The PKI module received the CA certificate through SCEP.
Before pressing ENTER you must choose 'YES' or 'NO'[Y/N]:y
Saving CA/RA certificates chain, please wait a moment......
CA certificates retrieval success.
[Sysname]
domain crt is trusted.
// The CA root certificate of PKI domain crt is trusted.
The PKI module updated the CA certificate of PKI domain crt successfully.

The PKI module obtained the CA certificate of PKI domain crt successfully.

The PKI module obtained the first certificate of the certificate chain.

The PKI module added the certificate to the trusted certificate list.

The PKI module sent a local certificate request.

Certificate is being requested, please wait......

Enrolling the local certificate, please wait a while......

The SCEP ID is 8DDAA53A978FB25EDBC614561176E98D.

The size of the certificate request message in PKCS#10 format is 231 bytes.

The payload has been encrypted successfully.

The size of the inner payload is 533 bytes.

The PKI module was creating the external PKCS#7 file.

The PKI module added the signature successfully.

The PKI module added string attributes transID and messageType and octet attribute senderNonce.

The PKI module encoded 1735 bytes of payload using base64 and encapsulated the certificate request message in PKCS#7 format.
Saving the local certificate to device......
Done!

// The local certificate was successfully enrolled and saved to the device.
%Aug  8 11:24:10:407 2006 Sysname PKI/4/Local_Cert_Request:Request local certificate of
the domain crt successfully.
*0.529407 Sysname PKI/7/PKI_Debug:SCEP receive message: Server returned status code 200

// The PKI module received a response from the CA server.
*0.529407 Sysname PKI/7/PKI_Debug:PKCS#7 develope: reading outer PKCS#7
*0.529422 Sysname PKI/7/PKI_Debug:PKCS#7 develope: PKCS#7 payload size: 2302 bytes
*0.529422 Sysname PKI/7/PKI_Debug:PKCS#7 develope: PKCS#7 contains 1254 bytes of enveloped
data
//The PKCS#7 file contains 1254 bytes of encapsulated data.
*0.529422 Sysname PKI/7/PKI_Debug:PKCS#7 develope: verifying signature
*0.529438 Sysname PKI/7/PKI_Debug:PKCS#7 develope: signature ok

// The signature is OK.
*0.529438 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding signed attributes
*0.529438 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute transId
*0.529454 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 32 bytes for attribute
*0.529454 Sysname PKI/7/PKI_Debug:PKCS#7 develope: reply transaction id:
8DDAA5A978FB25EDBC614561176E98D

// The PKI module found signed attribute transId and allocated 32 bytes for it.
*0.529469 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute messageType
*0.529469 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 1 bytes for attribute
*0.529469 Sysname PKI/7/PKI_Debug:PKCS#7 develope: reply message type is good

// The PKI module found attribute messageType and allocated one byte for it.
*0.529485 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute senderNonce
*0.529485 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 16 bytes for attribute
*0.529485 Sysname PKI/7/PKI_Debug:
PKCS#7 develope: senderNonce in reply: :
*0.529500 Sysname PKI/7/PKI_Debug:32e82738 c9383260 b6816ac2 b26671c6

// The PKI module found attribute senderNonce in the response and allocated 16 bytes for it.
*0.529500 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute recipientNonce
*0.529500 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 16 bytes for attribute
*0.529516 Sysname PKI/7/PKI_Debug:
PKCS#7 develope: recipientNonce in reply:
*0.529532 Sysname PKI/7/PKI_Debug:3673eedb 03ebe0fc 9c7d89ff 55eafba5

// The PKI module found attribute recipientNonce in the response and allocated 16 bytes for it.
*0.529532 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute pkiStatus
*0.529532 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 1 bytes for attribute
*0.529532 Sysname PKI/7/PKI_Debug:PKCS#7 develope: pkistatus SUCCESS

// The PKI module found attribute pkiStatus and allocated one byte for it.
*0.529547 Sysname PKI/7/PKI_Debug:PKCS#7 develope: reading inner PKCS#7
*0.529547 Sysname PKI/7/PKI_Debug:PKCS#7 develope: decrypting inner PKCS#7
*0.529547 Sysname PKI/7/PKI_Debug:PKCS#7 develope: PKCS#7 payload size: 1036 bytes
The PKI module de-encapsulated the PKCS#7 file and found that the payload size is 1036 bytes.

Enable certificate request debugging. Output similar to the following example is generated:

```
[Sysname] pki retrieval-certificate ca domain crt
```

Retrieving the CA certificate.

```
Retrieving CA/RA certificates. Please wait a while......
```

The PKI module sent a packet to the CA through SCEP.

```
The trusted CA's fingerprint is:
MD5  fingerprint:8FDC C669 7A95 5505 8C0A 8633 818D A0A1
SHA1 fingerprint:8CCD 07AD 6C9A 229B 3378 2430 F038 A142 D175 190E
```

The certificate fingerprint of the CA was calculated by using the hash algorithm.

```
Is the fingerprint correct?(Y/N):
```

The PKI module issued a local certificate request.

```
Certificate is being requested, please wait......
```

The PKI module created the DN used for certificate request.

```
```

The output shows the following:

- The host address of the CA server is 4.4.4.133.
- The port number of the SCEP protocol is 446.
- The path of the CA server is 6953bf7fb5b1cf514376243ce67ebed1209c292a.
- The SCEP ID is 038DE310FA1404F6781ED659158FE236.

*0.244438 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: creating inner PKCS#7
  // The PKI module was creating inner PKCS#7 file.
*0.244438 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: pkcs10 request 231 bytes
  // The size of the certificate request message in PKCS#10 format is 231 bytes.
*0.244438 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: successfully encrypted payload
  // The payload has been encrypted successfully.
*0.244454 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: inner payload size 533 bytes
  // The size of the inner payload is 533 bytes.
*0.244454 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: creating outer PKCS#7
  // The PKI module was creating the external PKCS#7 file.
*0.244454 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: signature added successfully
  // The PKI module added the signature successfully.
*0.244469 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: adding signed attributes
  *0.244469 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: adding string attribute transId
  *0.244469 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: adding string attribute messageType
  *0.244469 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: adding octet attribute senderNonce
  // The PKI module added string attributes transID and messageType and octet attribute senderNonce.
*0.244485 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: PKCS#7 data written successfully
  // The PKCS#7 file has been written into the memory successfully.
*0.244485 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: applying base64 encoding
  *0.244485 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: base64 encoded payload size: 1735 bytes
  // The PKI module encoded 1735 bytes of payload using base64 and encapsulated the certificate request message in PKCS#7 format.
*0.244516 Sysname PKI/7/PKI_Debug:SCEP send message:IP = 0x85040404
  // The PKI module sent a certificate request message through SCEP.
Certificate enroll Successfully!
Saving the local certificate to device......
Done!

%Aug  8 11:19:26:47 2006 Sysname PKI/4/Local_Cert_Request:Request local certificate of the domain crt successfully.
*0.245063 Sysname PKI/7/PKI_Debug:SCEP receive message: Server returned status code 200
  // The PKI module received a response from the CA server through SCEP.
*0.245063 Sysname PKI/7/PKI_Debug:Valid response from server
*0.245063 Sysname PKI/7/PKI_Debug:PKCS#7 develop: reading outer PKCS#7
  *0.245063 Sysname PKI/7/PKI_Debug:PKCS#7 develop: PKCS#7 payload size: 2302 bytes
  *0.245079 Sysname PKI/7/PKI_Debug:PKCS#7 develop: PKCS#7 contains 1254 bytes of enveloped data
  // The PKCS#7 file contains 1254 bytes of encapsulated data.
*0.245079 Sysname PKI/7/PKI_Debug:PKCS#7 develop: verifying signature
  *0.245079 Sysname PKI/7/PKI_Debug:PKCS#7 develop: signature ok
The signature is OK.
*0.245079 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding signed attributes
*0.245079 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute transId
*0.245079 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 32 bytes for attribute
*0.245094 Sysname PKI/7/PKI_Debug:PKCS#7 develope: reply transaction id: 038DE310FA1404F6781ED659158FE236

The PKI module found signed attribute transId and allocated 32 bytes for it.
*0.245094 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute messageType
*0.245094 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 1 bytes for attribute
*0.245094 Sysname PKI/7/PKI_Debug:PKCS#7 develope: reply message type is good

The PKI module found attribute messageType and allocated one byte for it.
*0.245110 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute senderNonce
*0.245110 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 16 bytes for attribute
*0.245110 Sysname PKI/7/PKI_Debug:PKCS#7 develope: senderNonce in reply:
*0.245125 Sysname PKI/7/PKI_Debug:PKCS#7 develope: 8fe84efc c3aa2942 ab0913b9 e6af539a

The PKI module found attribute senderNonce in the response and allocated 16 bytes for it.
*0.245125 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute recipientNonce
*0.245125 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 16 bytes for attribute
*0.245141 Sysname PKI/7/PKI_Debug:PKCS#7 develope: recipientNonce in reply:
*0.245141 Sysname PKI/7/PKI_Debug:cfbbbb7b a8737110 a5d618dd fb9133de

The PKI module found attribute recipientNonce in the response and allocated 16 bytes for it.
*0.245141 Sysname PKI/7/PKI_Debug:PKCS#7 develope: finding attribute pkiStatus
*0.245141 Sysname PKI/7/PKI_Debug:PKCS#7 develope: allocating 1 bytes for attribute
*0.245141 Sysname PKI/7/PKI_Debug:PKCS#7 develope: pkistatus SUCCESS

The PKI module found attribute pkiStatus and allocated one byte for it.
*0.245141 Sysname PKI/7/PKI_Debug:PKCS#7 develope: reading inner PKCS#7
*0.245141 Sysname PKI/7/PKI_Debug:PKCS#7 develope: decrypting inner PKCS#7
*0.245141 Sysname PKI/7/PKI_Debug:PKCS#7 develope: PKCS#7 payload size: 1037 bytes

The PKI module decapsulated the PKCS#7 file and found that the payload size is 1037 bytes.
*0.245141 Sysname PKI/7/PKI_Debug:PKI Get the Signed Certificates:
  subject: CN=crtSysname1
*0.245141 Sysname PKI/7/PKI_Debug: issuer: CN=sec,OU=software,O=aaa,C=cn
*0.245157 Sysname PKI/7/PKI_Debug: Key usage: general purpose

The PKI module obtained a certificate.
  - The DN of the entry is CN = crtSysname1.
  - The DN of the issuer is CN = sec,OU = software,O = aaa,C = cn.

Enable certificate validation debugging. When a certificate validation operation is performed, output similar to the following example is generated:

```
[Sysname] pki validate-certificate local domain crt
```

Validate the local certificate.

Verifying certificate......
  Serial Number:
The serial number of the certificate is FEADA5CA 028289AB CE95C6B6 E687639D.

Issuer:
   C=cn
   O=aaa
   OU=software
   CN=sec

The above shows the DN of the issuer.

Subject:
   CN=crtSysname3

The above shows the Entity DN of the certificate being validated.

*0.745641 Sysname PKI/7/PKI_Debug:CN=crtSysname3
*0.745657 Sysname PKI/7/PKI_Debug:error at 0 depth:subject issuer mismatch
*0.745657 Sysname PKI/7/PKI_Debug:Check the last certificate self signed.
*0.745657 Sysname PKI/7/PKI_Debug:CN=crtSysname3
*0.745657 Sysname PKI/7/PKI_Debug:error at 0 depth:subject issuer mismatch
*0.745672 Sysname PKI/7/PKI_Debug:Lookup certificate issuers and push into chain
*0.745672 Sysname PKI/7/PKI_Debug:CN=crtSysname3
*0.745672 Sysname PKI/7/PKI_Debug:error at 0 depth:subject issuer mismatch
*0.745688 Sysname PKI/7/PKI_Debug:Check certificates purpose.
*0.745688 Sysname PKI/7/PKI_Debug:Check certificates trust.
*0.746704 Sysname PKI/7/PKI_Debug:Check certificates revocation status.
Verify result: ok

The PKI module checked the purpose, signature, and revocation status of the certificate. All items are correct.


The PKI module updated and retrieved the CRL successfully.

*0.745782 Sysname PKI/7/PKI_Debug:ok
*0.745782 Sysname PKI/7/PKI_Debug:Verify certificate chain.

The local certificate passed validation.

%Sysname% pki validate-certificate ca domain crt
Verifying certificate......
Serial Number:
   EF425C97 E737B289 BB576A4B 5F657DB6
Issuer:
   C=cn
   O=hhh
   OU=software
   CN=sec
Subject:
   C=cn
Verifying certificate CN=sec, OU=software, O=hhh, C=cn of the domain crt successfully.

// The CA certificate passed validation.

# Enable PKI error debugging. When an incorrect URL is specified for certificate retrieval, output similar to the following example is generated:

<Sysname> debugging pki error
[Sysname] pki retrieval-certificate ca domain crt
Retrieving CA/RA certificates. Please wait a while......
Get CA/RA certificates error.

[Sysname]
%Aug 21 10:35:15:766 2006 Sysname PKI/7/PKI_Debug:SCEP receive message: Server returned status code 200
%Aug 21 10:35:15:766 2006 Sysname PKI/7/PKI_Debug:SCEP receive message: wrong MIME content type
%Aug 21 10:35:15:766 2006 Sysname PKI/7/PKI_Debug:Error while sending message

// The SCEP message received from the CA states that the URL is incorrect.

[Sysname-pki-domain-crt] display this
# pki domain crt
  ca identifier rsaSysname
  certificate request url http://4.4.4.133:446/6953bf7f5b1cf514376243ce67ebed12
  certificate request from ca
  certificate request entity crt
  certificate request polling interval 5
certificate request polling count 5
  certificate request mode auto password simple kkk
  crl url http://4.4.4.133:447/security%20rsa.crl
  ldap-server ip 4.4.4.133
# return
[Sysname-pki-domain-crt] certificate request url
http://4.4.4.133:446/6953bf7f5b1cf514376243ce67ebed1209c292a
[Sysname-pki-domain-crt] quit
[Sysname] pki retrieval-certificate ca domain crt
Retrieving CA/RA certificates. Please wait a while......
The trusted CA's finger print is:

MD5 fingerprint: 8FDCC669 7A95 5505 8C0A 8633 818D A0A1
SHA1 fingerprint: 8CCD07AD 6C9A 229B 3378 2430 F038 A142 D175 190E

Is the finger print correct?(Y/N):

Saving CA/RA certificates chain, please wait a moment......
CA certificates retrieval success.

// The PKI module obtained the CA certificate successfully.

[Sysname]


// The CA root certificate of PKI domain crt is trusted.


// The PKI module updated the CA certificate of PKI domain crt successfully.


// The PKI module obtained the CA certificate of PKI domain crt successfully.

Certificate is being requested, please wait......

[Sysname]
Enrolling the local certificate,please wait a while......

Certificate enroll failed!

// The payload was encrypted successfully.

%Aug 21 10:37:26:946 2006 Sysname PKI/4/Local_Cert_Request:Fail to request local certificate of the domain crt.

// The PKI module failed to request a local certificate for PKI domain crt.


// The status code returned by the CA server is 200.


*Aug 21 10:37:26:962 2006 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: allocating 1 bytes for attribute


*Aug 21 10:37:26:977 2006 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: allocating 16 bytes for attribute

*Aug 21 10:37:26:977 2006 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: allocating 1 bytes for attribute

*Aug 21 10:37:26:977 2006 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: pkistatus FAILURE

*Aug 21 10:37:26:993 2006 Sysname PKI/7/PKI_Debug:PKCS#7 envelope: allocating 1 bytes for attribute
*Aug 21 10:37:26:993 2006 Sysname PKI/7/PKI_Debug:PKCS#7 develope: reason: Transaction not permitted or supported
*Aug 21 10:37:26:993 2006 Sysname PKI/7/PKI_Debug:PKCS#7 develope: illegal size of payload

// The size of the payload after PKCS#7 decapsulation is invalid.
*Aug 21 10:37:27:16 2006 Sysname PKI/7/PKI_Debug:Certificate enroll failed, error code is 72

// The PKI module failed to request a local certificate because the key pair was already used.
**Policy-based routing debugging commands**

The output description tables in this document only contain fields and messages that require an explanation.

**debugging ip policy-based-route**

Use **debugging ip policy-based-route** to enable policy-based routing (PBR) debugging. Use **undo debugging ip policy-based-route** to disable PBR debugging.

**Syntax**

```
debugging ip policy-based-route
undo debugging ip policy-based-route
```

**Default**

PBR debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Usage guidelines**

Table 1 describes output fields and messages for the **debugging ip policy-based-route** command.

**Table 134 Output from the debugging ip policy-based-route command**

<table>
<thead>
<tr>
<th>a. Field</th>
<th>b. Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c. DP</td>
<td>d. Data plane.</td>
</tr>
<tr>
<td></td>
<td>e. This field is only supported by multi-core devices.</td>
</tr>
<tr>
<td>f. IP policy based routing success</td>
<td>g. PBR succeeded.</td>
</tr>
<tr>
<td>h. IP policy based routing fail for interface index [STRING] not exist</td>
<td>i. PBR failed because the interface with index [STRING] does not exist.</td>
</tr>
<tr>
<td></td>
<td>j. Support for this output depends on the device model.</td>
</tr>
<tr>
<td>k. IP policy based routing fail for next-hop [STRING] error</td>
<td>l. PBR failed because the next hop with index [STRING] does not exist.</td>
</tr>
<tr>
<td>m. No memory space while [STRING]</td>
<td>n. Memory allocation for [STRING] failed due to insufficient memory space.</td>
</tr>
<tr>
<td>o. headlen</td>
<td>p. Header length.</td>
</tr>
<tr>
<td>a. Field</td>
<td>b. Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>q. tos</td>
<td>r. Type of service.</td>
</tr>
<tr>
<td>s. pktlen</td>
<td>t. Packet length.</td>
</tr>
<tr>
<td>u. pktid</td>
<td>v. Packet ID.</td>
</tr>
<tr>
<td>w. protocol</td>
<td>x. Upper layer protocol.</td>
</tr>
<tr>
<td>y. s</td>
<td>z. Source address.</td>
</tr>
<tr>
<td>aa. d</td>
<td>bb. Destination address.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable PBR debugging. The output in this example was created when the following conditions exist:

- Configure PBR on Ethernet 1/0 to allow forwarding packets with a length from 10 to 100 bytes to Serial 2/0.
- On another device, send ping packets (with a length from 10 to 100 bytes) to Ethernet 1/0.

```bash
<Sysname> debugging ip policy-based-route
*0.513578 Sysname PBR/8/POLICY-ROUTING:IP policy based routing success : output interface Serial2/0 is down
// PBR succeeded, but the outgoing interface Serial 2/0 was down, resulting in packet forwarding failures.
```
Port security debugging commands

The port security module name is identified as "PORTSEC" in debugging messages.

Some information in this chapter is device type specific. Devices in this chapter are categorized depending on their IRF capability and support for interface cards that use independent processors for forwarding traffic, as shown in Table 1.

Table 135 Device types

<table>
<thead>
<tr>
<th>Device type</th>
<th>Interface cards with on-card processors</th>
<th>IRF capability</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed devices</td>
<td>Yes</td>
<td>No</td>
<td>HP 6600 routers (except for 6602)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes (in standalone mode)</td>
<td>HP 12500 switches \ HP 10500 switches</td>
</tr>
<tr>
<td>Distributed IRF devices</td>
<td>Yes</td>
<td>Yes (in IRF mode)</td>
<td>HP 12500 switches \ HP 10500 switches</td>
</tr>
<tr>
<td>Centralized devices</td>
<td>No</td>
<td>No</td>
<td>HP MSR routers \ HP 6602 router</td>
</tr>
<tr>
<td>Centralized IRF devices</td>
<td>No</td>
<td>Yes</td>
<td>HP 5800 switches \ HP 5500 switches</td>
</tr>
</tbody>
</table>

The output description tables in this document only contain fields and messages that require an explanation.

**debugging port-security**

Use `debugging port-security` to enable port security debugging.

Use `undo debugging port-security` to disable port security debugging.

**Syntax**

Centralized devices:

```
debugging port-security { all | error | event }
undo debugging port-security { all | error | event }
```

Distributed devices/centralized IRF devices:

```
debugging port-security { all | error | event } [ slot slot-number ]
undo debugging port-security { all | error | event } [ slot slot-number ]
```

Distributed IRF devices:

```
debugging port-security { all | error | event } [ chassis chassis-number slot slot-number ]
undo debugging port-security { all | error | event } [ chassis chassis-number slot slot-number ]
```
Default

Port security debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all: Specifies all types of port security debugging.

error: Specifies port security error debugging.

event: Specifies port security event debugging.

slot slot-number: Specifies a card by its slot number. (Distributed devices—In standalone mode.)

slot slot-number: Specifies the ID of an IRF member device. (Centralized IRF devices.)

chassis chassis-number slot slot-number: Specifies a card on an IRF member device. The chassis-number argument represents the member ID of the IRF member device. The slot-number argument represents the slot number of the card. (Distributed devices—In IRF mode.)

Usage guidelines

Table 2 describes output fields and messages for the `debug port-security error` command.

Table 136 Output from the debugging port-security error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidAuth</td>
<td>Invalid user.</td>
</tr>
<tr>
<td>Failed to handle set operation of PortSec security MAC table.</td>
<td>The port security module failed to configure secure MAC entries.</td>
</tr>
<tr>
<td>Failed to set PortMode.</td>
<td>The port security module failed to set port security mode.</td>
</tr>
<tr>
<td>Failed to get physical control of IfIndex.</td>
<td>The port security module failed to obtain the physical layer control information of the specified interface.</td>
</tr>
<tr>
<td>Configure IntrusionAction when failed to get portCB.</td>
<td>The port security module set intrusion protection action when it failed to obtain the port control block.</td>
</tr>
<tr>
<td>Error: Unable to delete any MACs at interface: IfIndex. Self slot is SlotNum.</td>
<td>The port security module failed to delete any MAC address at interface IfIndex. The slot number is SlotNum.</td>
</tr>
<tr>
<td>Failed to set PortCB by handle.</td>
<td>The device failed to configure the port data structure on the port security control block.</td>
</tr>
<tr>
<td>Failed to delete security MAC from list. Self slot is SlotID.</td>
<td>The port security module failed to delete secure MAC entries from the table. The slot is SlotID.</td>
</tr>
<tr>
<td>There is no security MAC match the delete condition. Self slot is SlotID.</td>
<td>No secure MAC entry matches the delete condition. The slot is SlotID.</td>
</tr>
<tr>
<td>Port Security’s state is disabled!</td>
<td>Port security is not enabled.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Failed to Alloc memory for RPC message, when request backing up</td>
<td>The device failed to allocate memory for RPC messages when it was backing up the MAC address table to other cards except the active MPU.</td>
</tr>
<tr>
<td>MAC Address table to slave board!</td>
<td>(Distributed devices/distributed IRF devices.)</td>
</tr>
<tr>
<td></td>
<td>The device failed to allocate memory for RPC messages when it was backing up the MAC address table to the subordinate devices. (Centralized IRF devices.)</td>
</tr>
<tr>
<td>Failed to alloc memory for RPC message, when request backing up</td>
<td>The device failed to allocate memory for RPC messages when it was backing up the blocked MAC address table to other cards except the active MPU.</td>
</tr>
<tr>
<td>BlockMac table to slave board.</td>
<td>(Distributed devices/distributed IRF devices.)</td>
</tr>
<tr>
<td></td>
<td>The device failed to allocate memory for RPC messages when it was backing up the blocked MAC address table to the subordinate devices. (Centralized IRF devices.)</td>
</tr>
<tr>
<td>RPC failed, when backup BlockMac table to new board! (RPC wrong=</td>
<td>The device failed to allocate memory for RPC messages when it was backing up the blocked MAC address table to the new card. (Distributed devices/distributed IRF devices.)</td>
</tr>
<tr>
<td>Error Code)</td>
<td>The device failed to allocate memory for RPC messages when it was backing up the blocked MAC address table to the new subordinate device. (Centralized IRF devices.)</td>
</tr>
<tr>
<td>RPC failed, the BlockMac table is not synchronized to the new board.</td>
<td>RPC failed and the secure MAC address table was not synchronized to the new card. (Distributed devices/distributed IRF devices.)</td>
</tr>
<tr>
<td></td>
<td>RPC failed and the secure MAC address table was not synchronized to the new subordinate device. (Centralized IRF devices.)</td>
</tr>
<tr>
<td>Failed to show port.</td>
<td>The port security module failed to display port information.</td>
</tr>
<tr>
<td>Failed to show BlockMac.</td>
<td>The port security module failed to display blocked MAC entries.</td>
</tr>
<tr>
<td>Failed to sync security MAC for distribution.</td>
<td>The port security module failed to synchronize the secure MAC entries to other cards. (Distributed devices/distributed IRF devices.)</td>
</tr>
<tr>
<td></td>
<td>The port security module failed to synchronize the secure MAC entries to other devices. (Centralized IRF devices.)</td>
</tr>
<tr>
<td>Failed to malloc for RPC message, when request getting portsec-</td>
<td>The port security module failed to allocate memory for RPC messages when it tried to obtain port security information from an interface card.</td>
</tr>
<tr>
<td>security port information from IO board.</td>
<td>(Distributed devices/distributed IRF devices.)</td>
</tr>
</tbody>
</table>
| Failed to operate get-next-BlockMac:BlockMac list reached end.       | The port security module failed to perform the get-next-BlockMac operation. The current entry is the last entry in the BlockMac list.
Failed to malloc for RPC message, when request getting BlockMac number from a board.
The device failed to allocate memory for RPC messages when it tried to obtain the number of blocked MAC addresses.

Failed to malloc for RPC message, when request getting BlockMac address from a board.
The device failed to allocate memory for RPC messages when it tried to obtain blocked MAC addresses.

MIB:OUITable proc block number is wrong.
MIB operation of OUI MAC entries. The number of message blocks is incorrect.

MIB OUI set: self slot is in slave status, backup auto.
The port security module set the OUI table through a MIB operation. The current MPU is a standby MPU and backs up the OUIs automatically. (Distributed devices/distributed IRF devices.) The current device is a subordinate device and backs up the OUIs automatically. (Centralized IRF devices.)

Table 3 describes output fields and messages for the `debugging port-security event` command.

### Table 137 Output from the debugging port-security event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidAuth</td>
<td>Invalid user.</td>
</tr>
<tr>
<td>Portsec SecureMAC table process begin...</td>
<td>The port security module began to add secure MAC entries.</td>
</tr>
<tr>
<td>Configure IntrusionAction, the same as current config.</td>
<td>The Intrusion action to be configured is the same as the action that is already configured.</td>
</tr>
<tr>
<td>Configure IntrusionAction, NoShutdown the port.</td>
<td>The port security module brought up the port, because the port has been shut down by the intrusion protection feature.</td>
</tr>
<tr>
<td>Configure IntrusionAction, port is not shutdown yet.</td>
<td>Intrusion protection action is configured and the port is not shut down.</td>
</tr>
<tr>
<td>PortSec NoShutdown port...</td>
<td>The port security module brought up the port.</td>
</tr>
<tr>
<td>PortSec receives a new SrcUnkown MAC but portmode is not autolern, secure or userloginwithoui, bypass it to macauth !</td>
<td>The port received a packet with an unknown source MAC, but the port security mode is not autoLearn, secure, or userLoginWithOUI. The port security module starts to perform MAC authentication.</td>
</tr>
<tr>
<td>Get NULL pointer for delete count, alloc a memory block for it...</td>
<td>The port security module prepares to remove the secure MAC entries. Because the count pointer is null, the device will allocate a memory block for it.</td>
</tr>
<tr>
<td>Security MAC distributed-backup is finished on slot SlotID.</td>
<td>The port security module finished adding secure MAC entries on the card in slot SlotID. (Distributed devices/distributed IRF devices.) The port security module finished adding secure MAC entries on device SlotID. (Centralized IRF devices.)</td>
</tr>
</tbody>
</table>
### Field Description

- **Broadcast send a MAC address sync msg,**
  - IfIndex: IFINDEX
  - Status : STATUS
  - VlanID : VLANID
  - Mac-addr: MacAddress
  - OpType: OPTYPE
  - Self slot is SELFSLOTID.

  The port security module broadcast a MAC address synchronization message.

- **RPC receive show port msg from self board!**

  The port security module received an RPC message for displaying port information of the current card.
  (Distributed devices/distributed IRF devices.)

  The port security module received an RPC message for displaying port information of the current device.
  (Centralized devices/centralized IRF devices.)

---

### Examples

# Enable port security error debugging. Output similar to the following example is generated when you perform the following tasks:

- Enable port security on the device.
- Change the port security mode to secure.
- Add a secure MAC address.

```bash
<Sysname> debugging port-security error
<Sysname> port-security enable
Notice: The port-control of 802.1X will be restricted to auto when port-security is enabled.
   Please wait...
Done.
<Sysname> interface gigabitethernet 1/1
<Sysname-GigabitEthernet1/1] port-security port-mode secure
<Sysname-GigabitEthernet1/1] port-security mac-address security 1-1-1 vlan 1
Error: Can not operate security MAC address for current port mode is not autoLearn!
*Apr 28 23:54:57:979 2000 Sysname PORTSEC/7/Event:InvalidPort,InvalidAuth,error occured while adding security MAC(0001-0001-0001, IfIndex 0x900002) to list.
```

// The port security module failed to add the MAC address to the secure MAC address table, because the port security mode of the current port was not autoLearn.

# Enable port security event debugging. Output similar to the following example is generated when you perform the following tasks:

- Enable port security on the device.
- Change the port security mode to secure.
- Add a secure MAC address.

```bash
<Sysname> debugging port-security event
<Sysname> port-security enable
Notice: The port-control of 802.1X will be restricted to auto when port-security is enabled.
   Please wait...
Done.
<Sysname> interface gigabitethernet 1/1
<Sysname-GigabitEthernet1/1] port-security port-mode secure
<Sysname-GigabitEthernet1/1] port-security mac-address security 1-1-1 vlan 1
```

---

228
Error: Can not operate security MAC address for current port mode is not autoLearn!

*Apr 28 23:54:57:975 2000 Sysname PORTSEC/7/Event:InvalidPort,InvalidAuth,ExecID=1,Otype=2, Calling portsec cfg module...

// The device called the port security configuration module.

*Apr 28 23:54:57:976 2000 Sysname PORTSEC/7/Event:InvalidPort,InvalidAuth,CFG:Portsec SecureMAC table process begin...

// The port security module tried to add a secure MAC entry.

*Apr 28 23:54:57:978 2000 Sysname PORTSEC/7/Event:Port:GigabitEthernet1/1,InvalidAuth, PortMode is not autoLearn!

// The port security module failed to add the secure MAC entry because the port security mode was not autoLearn.
Portal debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging portal

Use **debugging portal** to enable portal debugging.

Use **undo debugging portal** to disable portal debugging.

**Syntax**

```
debugging portal { { acl | all | connection | packet [ acl acl-number ] } interface interface-type interface-number | error | tcp-cheat }
undo debugging portal { { acl | all | connection | packet } interface interface-type interface-number | error | tcp-cheat }
```

**Default**

Debugging for portal is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **acl**: Specifies portal ACL debugging.
- **all**: Specifies all types of portal debugging.
- **connection**: Specifies portal connection debugging.
- **packet**: Specifies portal packet debugging.
- **acl acl-number**: Specifies an ACL to filter portal packets. The value range for the acl-number argument is 2000 to 3999. If you specify ACLs multiple times, the most recently specified ACL applies. Only the source IP address field in the specified ACL rules is used for packet filtering.
- **interface**: Specifies debugging on an interface.
- **interface-type interface-number**: Specifies an interface by its type and number.
- **error**: Specifies portal error debugging.
- **tcp-cheat**: Specifies portal TCP anti-spoofing debugging.

**Usage guidelines**

Table 1 describes output fields and messages for the **debugging portal tcp-cheat** command.
Table 138 Output from the debugging portal tcp-cheat command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Source information flag.</td>
</tr>
<tr>
<td>MAC</td>
<td>Source MAC address of TCP connection.</td>
</tr>
<tr>
<td>VLAN</td>
<td>ID of Source VLAN of TCP connection.</td>
</tr>
<tr>
<td>PortIndex</td>
<td>Index of source port of TCP connection.</td>
</tr>
<tr>
<td>changed from STATUS_A to STATUS_B</td>
<td>The status is changed from status A to status B.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the `debugging portal packet` command.

Table 139 Output from the debugging portal packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portal socket receive packet length: <code>packetlength</code></td>
<td>Length of the packet received through the portal socket.</td>
</tr>
<tr>
<td>Portal check packet OK</td>
<td>Indicates whether the received packet is legal.</td>
</tr>
<tr>
<td>Portal packet head</td>
<td>Packet header.</td>
</tr>
<tr>
<td>SN: <code>serialNo</code></td>
<td>Packet serial number.</td>
</tr>
<tr>
<td>Type: <code>PacketType</code></td>
<td>Packet type.</td>
</tr>
<tr>
<td>AttrNum: <code>Number</code></td>
<td>Number of attributes of the packet.</td>
</tr>
<tr>
<td>ErrCode: <code>Number</code></td>
<td>Packet error code.</td>
</tr>
<tr>
<td>UserIP: <code>IP-Address</code></td>
<td>IP address of the accessing user.</td>
</tr>
<tr>
<td>Portal packet attribute list</td>
<td>Attribute list of the packet.</td>
</tr>
<tr>
<td>Portal raw packet</td>
<td>Packet form.</td>
</tr>
<tr>
<td>Portal socket send packet length: <code>packetlength</code></td>
<td>Length of the packet sent through the portal socket.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the `debugging portal connection` command.

Table 140 Output from the debugging portal connection command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer <code>timer-name</code> is started</td>
<td>Started timer named <code>timer-name</code>.</td>
</tr>
<tr>
<td>user index</td>
<td>User index.</td>
</tr>
<tr>
<td>state</td>
<td>Current state of the user.</td>
</tr>
<tr>
<td>IP</td>
<td>IP address of the user.</td>
</tr>
<tr>
<td>Timer <code>timer-name</code> is stopped</td>
<td>A timer named <code>timer-name</code> is stopped.</td>
</tr>
<tr>
<td>send message-name message to moduleA</td>
<td>Sent a message named <code>message-name</code> to module A.</td>
</tr>
<tr>
<td>State <code>stateA</code> changing to <code>stateB</code></td>
<td>User state changed from state A to state B.</td>
</tr>
<tr>
<td>Recv portal message-name in state-name state</td>
<td>Received a message named <code>message-name</code> in the state named <code>state-name</code>.</td>
</tr>
<tr>
<td>[ACM]</td>
<td>Debugging information output by ACM module.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Processing message-name</td>
<td>Processing the message named message-name.</td>
</tr>
<tr>
<td>1 user index IP ip-address</td>
<td>User index is index, and IP address is ip-address.</td>
</tr>
<tr>
<td>recv message-name from ACM</td>
<td>Received a message named message-name from ACM module.</td>
</tr>
<tr>
<td>recv message-name from DRV</td>
<td>Received a message named message-name from the driver.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the debugging portal acl command.

**Table 141 Output from the debugging portal acl command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portal interface</td>
<td>Name of the interface enabled with Portal.</td>
</tr>
<tr>
<td>Status</td>
<td>Portal running status. “Running” means the portal is running normally.</td>
</tr>
<tr>
<td>Action</td>
<td>Action of the matching rule:</td>
</tr>
<tr>
<td></td>
<td>• permit.</td>
</tr>
<tr>
<td></td>
<td>• redirect.</td>
</tr>
<tr>
<td></td>
<td>• deny.</td>
</tr>
<tr>
<td></td>
<td>• NULL</td>
</tr>
<tr>
<td>Match rule</td>
<td>Matching rule.</td>
</tr>
<tr>
<td>Inbound interface</td>
<td>Inbound interface.</td>
</tr>
<tr>
<td>Type</td>
<td>Portal rule type: Static or Dynamic.</td>
</tr>
<tr>
<td>Source</td>
<td>Source information of the matching rule.</td>
</tr>
<tr>
<td>IP</td>
<td>Source IP address of the matching rule.</td>
</tr>
<tr>
<td>Mask</td>
<td>Source mask of the matching rule.</td>
</tr>
<tr>
<td>Protocol</td>
<td>Protocol number of the matching rule.</td>
</tr>
<tr>
<td>Mac</td>
<td>Source MAC address of the matching rule.</td>
</tr>
<tr>
<td>Interface</td>
<td>Source port number of the matching rule.</td>
</tr>
<tr>
<td></td>
<td>Source port number of the matching rule.</td>
</tr>
<tr>
<td></td>
<td>If Layer-2 interface is not bound, the value here is any for portal-enabled interfaces.</td>
</tr>
<tr>
<td>Vlan</td>
<td>Source VLAN ID of the matching rule.</td>
</tr>
<tr>
<td>Destination</td>
<td>Destination information of the matching rule.</td>
</tr>
<tr>
<td>IP</td>
<td>Destination IP address of the matching rule.</td>
</tr>
<tr>
<td>Mask</td>
<td>Destination mask of the matching rule.</td>
</tr>
<tr>
<td>TCP Port</td>
<td>Destination port number of the matching rule.</td>
</tr>
<tr>
<td>Context</td>
<td>Context sent by the driver module.</td>
</tr>
<tr>
<td>RuleID</td>
<td>Portal rule ID.</td>
</tr>
<tr>
<td>Sequence</td>
<td>Rule sequence number.</td>
</tr>
<tr>
<td>Failed to send core-message!</td>
<td>Failed to send inter-core messages.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>VCPU ID</td>
<td>Virtual CPU ID.</td>
</tr>
<tr>
<td>DPIfIndex</td>
<td>Data plane interface index.</td>
</tr>
<tr>
<td>SrcIP</td>
<td>Source IP address.</td>
</tr>
<tr>
<td>SrcMask</td>
<td>Source IP address mask.</td>
</tr>
<tr>
<td>SrcMac</td>
<td>Source MAC address.</td>
</tr>
<tr>
<td>DstIP</td>
<td>Destination IP address.</td>
</tr>
<tr>
<td>DstMask</td>
<td>Destination IP address mask.</td>
</tr>
<tr>
<td>Flow</td>
<td>Flow tag of packets.</td>
</tr>
<tr>
<td>AuthorACL</td>
<td>Authorization ACL number.</td>
</tr>
</tbody>
</table>

**Operation**
- **ADD**: Adds an ACL.
- **DEL**: Deletes an ACL.

| DstPort      | Destination port number.                        |

Table 5 describes output fields and messages for the **debugging portal error** command.

Table 142 **Output from the debugging portal error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User’s ip address did not match the configured auth-network. (user ip: ip)</td>
<td>User is not on any configured authentication subnet. (User IP: ip)</td>
</tr>
<tr>
<td>Can not find the interface for user access. (user ip: ip)</td>
<td>Layer 3 interface for user access was not found.</td>
</tr>
</tbody>
</table>

**Examples**

# On a portal-enabled device, enable portal TCP anti-spoofing debugging. When an unauthenticated portal user uses a browser to access a website, output similar to the following example is generated:

```bash
<Sysname> debugging portal tcp-cheat
*Dec 30 15:19:53:119 2007 Sysname TCPCHEAT/7/TCPCHEAT_DEBUG: Source MAC = 000d-88f8-0eab VLAN = 0, PortIndex = 0
45 00 00 30 86 b1 40 00
80 06 af 68 02 02 02 03
c0 a8 00 01 0b b4 00 50
6a a3 af d0 00 00 00 00
70 02 ff ff 97 f9 00 00
02 04 05 b4 01 01 04 02
// Port 0 of VLAN 0 received a TCP connection packet with source MAC address 000d-88f8-0eab.
```


*Dec 30 15:19:53:147 2007 Sysname TCPCHEAT/7/TCPCHEAT_DEBUG: Source MAC = 000d-88f8-0eab VLAN = 0, PortIndex = 0

// Portal added a connection with source IP 2.2.2.3 and source MAC address 000d-88f8-0eab. The connection accessed port 0 of VLAN 0.
// The status of the connection with source IP 2.2.2.3 is LISTEN.
*Dec 30 15:19:53:168 2007 Sysname TCPCHEAT/7/TCPCHEAT_DEBUG:State of connection with source IP 2.2.2.3 is LISTEN!
// The status of the connection with source IP 2.2.2.3 changed from LISTEN to SYN_RECV.
*Dec 30 15:19:53:177 2007 Sysname TCPCHEAT/7/TCPCHEAT_DEBUG:State of connection with source IP 2.2.2.3 changed from LISTEN to SYN_RECV!
// The status of the connection with source IP 2.2.2.3 changed from FIN_WAIT_1 to FIN_WAIT_2.
*Dec 30 15:19:53:507 2007 Sysname TCPCHEAT/7/TCPCHEAT_DEBUG:State of connection with source IP 2.2.2.3 changed from FIN_WAIT_1 to FIN_WAIT_2!
// The connection with source IP address 2.2.2.3 was removed.

# On a portal-enabled device, enable portal packet debugging. Perform portal authentication.
<Sysname> debugging portal packet interface ethernet 1/1
Portal socket receive packet length:34
Portal check packet OK
// Portal received a portal packet at socket.
Portal packet head:
SN:3234 Type:9 AttrNum:1 ErrorCode:0 UserIP:2.2.2.2
// The portal packet’s header information:
  • SN—sequence number
  • AttrNum—Number of attributes carried in the packet.
  • ErrorCode—Error code
  • UserIP—User IP address.
Portal packet attribute list:
  [  8 Port ] [  2 ] [ ]
// The attribute list carried in the packet: [ Attribute 8 Port ] [Length: 2] [Value: Null]
Portal raw packet:
  02 09 00 00 0c a2 00 00 02 02 02 00 00 00 01
  23 95 11 8b 1c 33 47 cf c5 1c de 8b bd 43 b8 24
  08 02
// The whole packet, displayed in hexadecimal notation.
*Dec 30 15:19:53:147 2007 Sysname PORTAL/8/PORTAL_DEBUG:
Portal socket send packet length:65
// Portal sent a portal packet through socket, with the length as 65 bytes
Portal packet head:
SN:3234 Type:10 AttrNum:2 ErrorCode:0 UserIP:2.2.2.2
// The portal packet’s header information:
  • SN—sequence number
  • AttrNum—Number of attributes carried in the packet.
  • ErrorCode—Error code
  • UserIP—User IP address.
Portal packet attribute list:
[  8 Port                ] [ 27] [AR49-45-vlan-00-0000@vlan]
[ 10 BAS-IP              ] [ 6] [2.2.2.1]

// The attribute list carried in the packet:
[Attribute 8 Port         ] [Length: 27] [Value: AR49-45-vlan-00-0000@vlan]
[Attribute 10 BAS-IP       ] [Length: 6] [Value: 2.2.2.1]

Portal raw packet:
02 0a 00 00 0c a2 00 00 02 02 02 02 00 00 00 02
dd 64 55 49 ee b3 4c 79 2a a4 9e 76 91 86 91 82
08 1b 41 52 34 39 2d 34 35 2d 76 6c 61 6e 2d 30
30 2d 30 30 30 30 40 76 6c 61 6e 0a 06 02 02 02
01

// The whole packet, displayed in hexadecimal notation.

# On a portal-enabled device, enable portal connection debugging. Perform portal authentication.
<Sysname> debug portal connection interface vlan 5
  Timer TMR_REQAUTH is started: user index 1 IP 24.24.0.2 state DISCOVERED!

// The TMR_REQAUTH timer was started for the user.
*Dec 30 15:19:53:147 2007 Sysname PORTAL/8/PORTAL_DEBUG:
  Timer TMR_REQAUTH is stopped: user index 1 IP 24.24.0.2 state DISCOVERED!

// The TMR_REQAUTH timer was stopped for the user.
  User: 1 IP: 24.24.0.2 state: DISCOVERED send AUTHREQ message to ACM!

// The user sent an AUTHREQ message to the ACM module.
  State DISCOVERED changing to WAIT_AUTHEN_ACK: user index 1 IP 24.24.0.2!

// The user's connection status changed from DISCOVERED to WAIT_AUTHEN_ACK.
  [ACM]User index 1: Recv portal AUTH-REQ in IDLE state!

// In the ACM module, the user with index 1 received an AUTHREQ message in IDLE state.
  [ACM]User index 1: Send AUTH-REQ to radius succ

// In the ACM module, the user with index 1 sent an AUTHREQ message to the RADIUS module successfully.
*Dec 30 15:19:53:197 2007 Sysname PORTAL/8/PORTAL_DEBUG:
  Processing AUTH-ACK user 1 IP 24.24.0.2 recv AUTH-ACCEPT from ACM!

// The user with IP address 24.24.0.2 and index 1 passed the authentication.
*0.1326010 Sysname PORTAL/8/PORTAL_DEBUG:
  Processing AUTH-ACK user 1 IP 24.24.0.2 recv ACL-SUCC from DRV!

// An ACL was successfully assigned to the user.
  [ACM]User index 1: Online Succ,Update user's online time!
// In the ACM module, the user with index 1 changed to ONLINE state. The user online time was refreshed.

# On a portal-enabled device, enable portal ACL debugging. Perform portal authentication. Output similar to the following example is generated when the device is with a single-core CPU:
<Sysname> debugging portal acl interface vlan 5
  Portal interface:Vlan-interface5
  Status:running, Action:redirect

// Portal was running normally on the portal-enabled interface VLAN-interface 5, which redirected the packet as instructed by the matching portal rule.
Match rule:
  Inbound interface = all
  Type = static
  Action = redirect

// The matching portal rule defines that the incoming interfaces are all interfaces, the rule type is static, and the action is redirect.
Source:
  IP = 0.0.0.0
  Mask = 0.0.0.0
  Protocol = 6
  MAC = 0000-0000-0000
  Interface = any
  VLAN = 2

// The source information of the portal rule.
Destination:
  IP = 0.0.0.0
  Mask = 0.0.0.0
  TCP Port = 80
  Context = 0x00000002,0xffffffff

// The destination information of the portal rule.
  Portal interface:Vlan-interface5
  Status:running, Action:permit

// Portal was running correctly on the portal-enabled VLAN-interface 5, which permitted the packet as instructed by the matching portal rule.
Match rule:
  Inbound interface = all
  Type = static
  Action = permit

// The matching portal rule information:
- **Inbound interface**—Incoming interfaces of the packets.
- **Type**—Type of the portal rule.
- **Action**—Action to be taken to the packets.
  Source:
    IP = 0.0.0.0
Mask = 0.0.0.0
Protocol = 0
MAC = 0000-0000-0000
Interface = any
VLAN = 2

// The source information of the portal rule.
Destination:
IP = 192.168.0.111
Mask = 255.255.255.255
TCP Port = 0
Context = 0x00000000,0xffffffff

// The destination information of the portal rule.
Inbound interface = all
Type = dynamic
Action = permit

// The upper layer software assigned a portal rule to the driver successfully.
Source:
IP = 24.24.0.2
Mask = 255.255.255.255
Protocol = 0
MAC = 0000-0000-0000
Interface = any
VLAN = 2

// The source information of the portal rule.
Destination:
IP = 0.0.0.0
Mask = 0.0.0.0
TCP Port = 0
Context = 0x0000000b,0xffffffff

// The destination information of the portal rule.
[Sysname] portal delete-user all
Type = dynamic
Action = permit

// The upper layer software removed a portal rule from the driver successfully.
Source:
IP = 24.24.0.2
Mask = 255.255.255.255
Protocol = 0
MAC = 0000-0000-0000
Interface = any
VLAN = 2

// The source information of the portal rule.
Destination:
The output in the following examples was created on portal-enabled devices with a multi-core CPU.

# Enable portal ACL debugging on the device. When a user pings the remote server, output similar to the following example is generated:

```
<Sysname> debugging portal acl interface gigabitethernet 1/1
Match Permit ACL.
DPIfIndex=49153, SrcIP=76.1.1.1, DstIP=192.168.0.244, Flow=87043375!
```

// Packets matched the permit rule.

# Enable portal ACL debugging on the device. When an authenticated portal user accesses the external IP address 192.168.0.222 through IE, output similar to the following example is generated:

```
<Sysname> debugging portal acl interface gigabitethernet 1/1
Match Permit ACL.
DPIfIndex=49153, SrcIP=76.1.1.1, DstIP=192.168.0.222, Flow=16777415!
```

// Packets matched the permit rule.

# Enable portal ACL debugging on the device. When a user failing portal authentication accesses the external IP address 192.168.0.251 through IE, output similar to the following example is generated:

```
<Sysname> debugging portal acl interface gigabitethernet 1/1
Match Redirect ACL.
DPIfIndex=49153, SrcIP=76.1.1.1, DstIP=192.168.0.251, Flow=16777417!
```

// Packets matched the redirect rule.

# Enable portal ACL debugging on the device. When a user failing portal authentication accesses the external IP address 192.168.0.221 through FTP, output similar to the following example is generated:

```
<Sysname> debugging portal acl interface gigabitethernet 1/1
Match Deny ACL. SrcIP=76.1.1.1, DstIP=192.168.0.221, Flow=26778415!
```

// Packets matched the deny rule.

# Enable portal ACL debugging on the device, and enable portal on interface GigabitEthernet 1/1.

```
<Sysname> debugging portal acl interface gigabitethernet 1/1
DRV_FUNC2:
  DPIfIndex = 49163;
  SrcIP = 0.0.0.0;
  SrcMask = 0.0.0.0;
  SrcMac = 0000-0000-0000;
  DstIP = 192.168.0.244;
  DstMask = 255.255.255.255;
  AuthorACL = 0xFFFFFFFF;
  Operation = ADD;
```

238
// The data plane received parameter information. The operation is to add a portal rule.

Added permit ACL: Successfully!
RuleID = 0x00000001
Sequence = 0x00000001

// The portal module successfully added a permit rule. Rule ID is 1, and rule sequence is 1.

Portal enabled on the interface!

// Portal was enabled on the interface.

DRV_FUNC3:
    DPIfIndex = 41963;
    SrcIP = 0.0.0.0;
    SrcMask = 0.0.0.0;
    DstPort = 80;
    Operation = ADD;

// The data plane received parameter information. The operation is to add a portal rule.

Added redirect ACL: Successfully!

// The portal module successfully added a redirect rule.

DRV_FUNC4:
    DPIfIndex = 41963;
    SrcIP = 0.0.0.0;
    SrcMask = 0.0.0.0;
    Operation = ADD;

// The data plane received parameter information. The operation is to add a portal rule.

Added deny ACL: Successfully!

// The portal module successfully added a deny rule.

# Enable portal authentication on an interface of the device. Enable portal error debugging. When online user (18.18.0.5) re-initiates portal authentication on the interface, output similar to the following example is generated:

<Sysname> debugging portal error

*Apr 26 14:50:55:032 2010 Sysname PORTAL/7/PORTAL_DEBUG: Received req-info: User already exists. (user ip: 18.18.0.5)

// The portal module received a req-info message. The user (IP: 18.18.0.5) is already online.

debbuging portal server

Use debugging portal server to enable portal server debugging.
Use undo debugging portal to disable portal server debugging.

Syntax

debugging portal server
undo debugging portal server

Default

Portal server debugging is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

Table 6 describes output fields and messages for the debugging portal server command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| User user-ip is being detected, index = index, detection mode = mode. | Detecting user user-ip. index indicates the user index. mode indicates the detection mode:  
  - 0: No detection.  
  - 1: Traffic detection.  
  - 2: ICMP packet detection. |
| Failed to write msg message to portal message queue. | Failed to write the msg message to the portal message queue. msg indicates the message type:  
  - Reqinfo: Request information.  
  - ReqAuth: Authentication request.  
  - ReqLogout: Logout request.  
  - AckNtfLogout: Force-logout request. |
| Received message from portal, IP = ip-address, type = type. | Received a message from the portal module. ip-address indicates the user IP. type indicates the message type:  
  - REACH_USER_LIMIT: Number of users has reached the maximum.  
  - ACCT_SUCCESS: Accounting succeeded.  
  - ACCT_FAIL: Accounting failed.  
  - PT_NORMAL_ACK_INFO: Information request response message.  
  - PT_NORMAL_NTF_LOGOUT: Force-logout message.  
  - PT_NORMAL_ACK_LOGOUT: Logout response message. |
| Received message from portal web server, IP = ip-address, type = type. | Received a message from the portal Web server. ip-address indicates the user IP. type indicates the message type:  
  - PT_NORMAL_REQ_INFO: Information request message.  
  - PT_NORMAL_REQ_AUTH: Authentication request message.  
  - PT_NORMAL_REQ_LOGOUT: Logout request message.  
<p>| Connection from WS is not for local server. | The Socket connection request received on the Web service (WS) module is not for the local server. |
| Failed to read data from connection. | Failed to read data from the Socket connection. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Received async connection is not for local server.</td>
<td>The received asynchronous Socket connection is not for the local server.</td>
</tr>
<tr>
<td>Failed to move connection from WS to portal.</td>
<td>Failed to move the Socket connection from the Web service module to the portal server.</td>
</tr>
<tr>
<td>Failed to write message to connection queue.</td>
<td>Failed to write data to the Socket connection queue.</td>
</tr>
<tr>
<td>Received invalid msg for connection.</td>
<td>Received a message indicating that the Socket connection is invalid.</td>
</tr>
<tr>
<td>Received connection from WS.</td>
<td>Received a Socket connection from the Web service module.</td>
</tr>
<tr>
<td>Connection is invalid.</td>
<td>The Socket connection is invalid.</td>
</tr>
<tr>
<td>Processing connection.</td>
<td>Processing the Socket connection.</td>
</tr>
<tr>
<td>Data from connection is not complete.</td>
<td>Data from the Socket connection is not complete.</td>
</tr>
<tr>
<td>Received close msg.</td>
<td>Received the Socket connection close message.</td>
</tr>
<tr>
<td>Saved connection.</td>
<td>The Socket connection was saved.</td>
</tr>
<tr>
<td>Deleted connection.</td>
<td>The Socket connection was deleted.</td>
</tr>
<tr>
<td>Conn ID: Reset age time to time.</td>
<td>The aging time of the specified connection (Conn ID) was set to time.</td>
</tr>
<tr>
<td>SSID ssidname bind File filename.</td>
<td>SSID is bound with a file.</td>
</tr>
<tr>
<td>Action: action, Type: type, UserName: username, Passwd: passwd.</td>
<td>URL information: request action, type, username, and password.</td>
</tr>
<tr>
<td>Action: POST, Type: LOGOFF.</td>
<td>Logoff request information.</td>
</tr>
<tr>
<td>Get HTTP data offset failed.</td>
<td>Failed to get HTTP data offset.</td>
</tr>
<tr>
<td>Cut HTTP header failed.</td>
<td>Failed to remove HTTP header.</td>
</tr>
<tr>
<td>Read HTTP data failed.</td>
<td>Failed to read HTTP data.</td>
</tr>
<tr>
<td>Parse HTTP data failed.</td>
<td>Failed to parse HTTP data.</td>
</tr>
<tr>
<td>Get HTTP parse ID failed.</td>
<td>Failed to get HTTP parse ID.</td>
</tr>
<tr>
<td>Set parse message type failed.</td>
<td>Failed to set the parse message type.</td>
</tr>
<tr>
<td>Write data failed.</td>
<td>Failed to write data.</td>
</tr>
<tr>
<td>Connection exists.</td>
<td>The Socket connection already exists.</td>
</tr>
<tr>
<td>Failed to malloc memory, cannot save connection.</td>
<td>Failed to allocate memory; cannot save the Socket connection.</td>
</tr>
<tr>
<td>Failed to get port or VLAN information (IP ip-addr).</td>
<td>Failed to get port or VLAN information of the user whose IP address is ip-addr.</td>
</tr>
<tr>
<td>Set MAC to zero (IP ip-addr).</td>
<td>The MAC address was set to 0 for the user whose IP address is ip-addr.</td>
</tr>
</tbody>
</table>
Failed to get MAC information (IP ip-addr).
Failed to get MAC information of the user whose IP address is ip-addr.

Examples

# On a device enabled with local portal server, enable portal server debugging. When a user uses the IE browser to trigger portal authentication, output similar to the following example is generated:

```<Sysname> debugging portal server
*Jan 10 16:17:01:94 2008 Sysname PORTAL/7/PORTAL_DEBUG: Received connection from WS
// The portal server processed the received portal connection request.
// The portal server replied with a logon file to the portal client, providing the logon page to the portal client.
*Jan 10 16:17:11:984 2008 Sysname PORTAL/7/PORTAL_DEBUG: Received read msg.
// The portal server received a logon post request from the portal client.
*Jan 10 16:17:11:984 2008 Sysname PORTAL/7/PORTAL_DEBUG: Received message from portal web server, IP = 2.2.20.4, type = PT_NORMAL_REQ_INFO.
// The portal module received an information request message from the local portal server.
*Jan 10 16:17:11:984 2008 Sysname PORTAL/7/PORTAL_DEBUG: Received message from portal, IP = 2.2.20.4, type = PT_NORMAL_ACK_INFO.
// The portal server received an information request response message from the portal module.
*Jan 10 16:17:12:187 2008 Sysname PORTAL/7/PORTAL_DEBUG: Received message from portal web server, IP = 2.2.20.4, type = PT_NORMAL_REQ_AUTH.
// The portal module received an authentication request message from the local portal server.
*Jan 10 16:17:12:390 2008 Sysname PORTAL/7/PORTAL_DEBUG: Received message from portal, IP = 2.2.20.4, type = PT_NORMAL_ACK_AUTH.
// The portal server received an authentication acknowledgement message from the portal module.
*Jan 10 16:17:12:390 2008 Sysname PORTAL/7/PORTAL_DEBUG: Sent web page to inform user 2.2.20.4: Authentication succeeded!
// The portal server successfully informed the portal client of authentication success.
*Jan 10 16:19:40:531 2008 Sysname PORTAL/7/PORTAL_DEBUG: User 2.2.20.4 is being detected, index = 0, detection mode = 2.
```
User 2.2.20.4 sent ICMP detection request packet.
User 2.2.20.4 received ICMP detection response packet.

// The portal server performed the ICMP detection.

Received connection 1 from WS.
Processing connection.
URL:POST, TYPE:LOGOFF.

// The portal server received a logoff request from the portal client.

Received message from portal web server, IP = 2.2.20.4, type = PT_NORMAL_REQ_LOGOUT.

// The portal module received a logoff request from the local portal server.

Received message from portal, IP = 2.2.20.4, type = PT_NORMAL_ACK_LOGOUT.
Replied a file logoffSuccess.htm for client.

// The local portal server informed the portal client of logoff success.
POS terminal access debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

documenting posa all

Use **debugging posa all** to enable all POS access debugging.

Use **undo debugging posa all** to disable all POS access debugging.

**Syntax**

```plaintext
debugging posa all [ terminal terminal-id | app app-id ]
undo debugging posa all [ terminal terminal-id | app app-id ]
```

**Default**

All POS access debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **terminal-id**: Specifies a POS terminal by its ID in the range of 1 to 255.
- **app-id**: Specifies a POS application by its ID in the range of 1 to 31.

**Usage guidelines**

For the command output description, see Table 1 and Table 2.

**Examples**

```plaintext
# Enable all types of POS access debugging. When a listening port is created on terminal 2 that already
# exists, output similar to the following example is generated:
<Sysname> debugging posa all
*Oct 18 16:23:19:486 2008 Sysname POSA/7/EVENT:
Terminal 2: Received ASYN_READ message from socket(12).
// Terminal 2 received an ASYN_READ message from the socket.
*Oct 18 16:23:19:486 2008 Sysname POSA/7/ERROR:
App 2: Failed to send flow packet, because the connect state(2) is not up.
// Application 2 failed to send the asynchronous packet, because the interface was not up.
*Oct 18 16:23:19:486 2008 Sysname POSA/7/PACKET:
Received 10 bytes from tcp terminal 2.
   PktLen(0x0008) ID(0x60) DST(0xaaa) SRC(0x1111)
   Total length: 10 Offset: 0, partial data as follows:
      0x000: 00 08 60 aa aa 11 11 11 11
// A 10-byte packet was received from TCP terminal 2.
```
debugging posa event

Use **debugging posa event** to enable POS access event debugging.

Use **undo debugging posa event** to disable POS access event debugging.

**Syntax**

```
debugging posa event [ terminal terminal-id | app app-id ]
undo debugging posa event [ terminal terminal-id | app app-id ]
```

**Default**

POS access event debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- `terminal-id`: Specifies a POS terminal by its ID in the range of 1 to 255.
- `app-id`: Specifies a POS application by its ID in the range of 1 to 31.

**Usage guidelines**

Table 1 describes output fields and messages for the **debugging posa event** command.

**Table 144 Output from the debugging posa event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>App n: Succeeded to send connecting request to server.</td>
<td>Application n successfully sent a connecting request to the FEP.</td>
</tr>
<tr>
<td>Terminal n: Received control message of deleting terminal.</td>
<td>POS access received a control message that terminal n is deleted.</td>
</tr>
<tr>
<td>App n: Received control message of disconnecting by app.</td>
<td>POS access received a control message that all connections to application n are terminated.</td>
</tr>
<tr>
<td>App n: Received control message of setting tcp keepalive option.</td>
<td>POS access received a control message that TCP keepalive parameters of application n are modified.</td>
</tr>
<tr>
<td>App n: Received line disable message.</td>
<td>POS access received a message that application n is switched between synchronous and asynchronous modes.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable POS access event debugging. When TCP application 2 in temporary connection mode is connected to asynchronous terminal 1, output similar to the following example is generated:

```
<Sysname> debugging posa event
*Jul 30 19:44:18:392 2008 posa POSA/7/EVENT:
```
debugging posa error

Use debugging posa error to enable POS access error debugging.
Use undo debugging posa error to disable POS access error debugging.

Syntax

debugging posa error [ terminal terminal-id | app app-id ]
undo debugging posa error [ terminal terminal-id | app app-id ]

Default

POS access error debugging is disabled.

Views

User view

Default command level

2: System level

Parameters

terminal-id: Specifies a POS terminal by its ID in the range of 1 to 255.
app-id: Specifies a POS application by its ID in the range of 1 to 31.

Usage guidelines

Table 2 describes output fields and messages for the debugging posa error command.

Table 145 Output from the debugging posa error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to create template of terminal(n), because of failing to malloc memory for terminal template.</td>
<td>POS access failed to create terminal template (n) because memory failed to be allocated for the template.</td>
</tr>
<tr>
<td>Terminal n: Failed to hung up fcm when reiniting terminal instance.</td>
<td>POS access failed to be hung up when terminal n was reset.</td>
</tr>
<tr>
<td>Failed to get correct terminal template by terminal(n) when creating listen socket.</td>
<td>POS access failed to obtain the correct terminal template through terminal (n) when it was creating the listen socket.</td>
</tr>
<tr>
<td>Failed to get app template by trans appID(n) when resetting terminal instance.</td>
<td>POS access failed to obtain the application terminal according to the transparent application ID (n) when it was resetting the terminal instance.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Failed to create terminal instance by terminal ((n)) when accepting connect from socket ((i)).</td>
<td>POS access failed to create a terminal instance through terminal ((n)) when a connection was received from socket ((i)).</td>
</tr>
<tr>
<td>Failed to set socket ((i)) to asyn mode when connecting to app.</td>
<td>POS access failed to set socket ((i)) to the asynchronous mode when a connection to application (n) was initiated.</td>
</tr>
<tr>
<td>App (n): Failed to create linking timer by socket ((i)).</td>
<td>POS access failed to create a timeout timer for application (n) through socket ((i)).</td>
</tr>
<tr>
<td>Failed to get terminal template by terminal ((n)) when updating the statistics.</td>
<td>POS access failed to obtain the terminal template through terminal ((n)) when packet statistics were refreshed.</td>
</tr>
<tr>
<td>Failed to unbind the application template ((n)) with interface ((m)) in command line.</td>
<td>POS access failed to unbind application template ((n)) from interface ((m)).</td>
</tr>
<tr>
<td>Failed to send message of creating listen socket when creating terminal template ((n)).</td>
<td>POS access failed to send the message that the listening port is created when it was creating terminal template (n).</td>
</tr>
<tr>
<td>Failed to create socket when checking terminal listen port ((n)).</td>
<td>POS access failed to create the socket when it was examining whether or not listening port (n) of the terminal is occupied.</td>
</tr>
<tr>
<td>Failed to bind listen port with socket checking terminal listen port ((n)).</td>
<td>POS access failed to bind the listening port to the socket when it was examining whether or not listening port (n) of the terminal is occupied.</td>
</tr>
<tr>
<td>Interface index ((n)) is invalid when binding application template ((m)) with interface.</td>
<td>Interface index ((n)) is invalid when the application template ((m)) is bound to the interface.</td>
</tr>
<tr>
<td>Failed to send message when binding interface ((n)) with application ((m)).</td>
<td>POS access failed to send the message that interface ((n)) is bound to application ((m)).</td>
</tr>
<tr>
<td>Failed to send message when setting tcp parameters of application template ((n)).</td>
<td>POS access failed to send the message that TCP parameters of application template (n) were set.</td>
</tr>
<tr>
<td>Failed to set fcm parameter of tradetime by mib.</td>
<td>POS access failed to set the FCM transaction time parameter through MIB.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
Failed to delete application (n) because of sending control message unsuccessfully. | POS access failed to delete application n because the control message requesting the deletion of the template failed to be sent.

Failed to create fcm terminal template in terminal table because of illegal interface index (n). | POS access failed to create the FCM terminal in the terminal table because the interface index n is invalid.

Failed to create flow terminal template because of illegal workmode on interface. | POS access failed to create the flow terminal because the interface worked in an improper mode.

Failed to remalloc memory when echoing global configuration. | POS access failed to reallocate memory when it was displaying global configuration information.

Failed to get template by ifindex(m) when echoing interface configuration. | POS access failed to obtain the template according to interface index m when it was displaying the interface configuration information.

Examples

# Enable POS access error debugging. When packets are received after the receive buffer of terminal 1 is full, output similar to the following example is generated:

```
<Sysname> debugging posa error
// POS access failed to add the asynchronous packet transferred in transparent mode to the receive buffer of terminal 1.
```

debugging posa packet

Use `debugging posa packet` to enable POS access packet debugging.

Use `undo debugging posa packet` to disable POS access packet debugging.

Syntax

```
debugging posa packet [ receive | send ] [ terminal terminal-id | app app-id ]
undo debugging posa packet [ receive | send ] [ terminal terminal-id | app app-id ]
```

Default

POS access packet debugging is disabled.

Views

User view

Default command level

2: System level
Parameters

**receive**: Specifies debugging for BPDU packets received.

**send**: Specifies debugging for BPDU packets sent.

**terminal-id**: Specifies a POS terminal by its ID in the range of 1 to 255.

**app-id**: Specifies a POS application by its ID in the range of 1 to 31.

Usage guidelines

Use the `receive` or `send`, and `terminal` or `app` keywords to enable or disable packet debugging in a specific direction on a specific POS terminal or application.

Special fields of the POS packet include the following:

- STX.
- PktLen (packet length).
- ID (ID of the Transport Protocol Data Unit ID, TPDU ID).
- DST (destination address of the TPDU).
- SRC (source address of the TPDU).
- EXT.
- CRC (cyclic redundancy check).

If a received packet is incomplete, fields that contain no data are displayed as 0. If a received packet is longer than 32 bytes, only the first 32 bytes are displayed.

Examples

```
# Enable POS access packet debugging. When Terminal 2 communicates with the POS terminal device in FCM mode, and is mapped to application 1 in TCP connection mode, output similar to the following example is generated:
<Sysname> debugging posa packet
*May 13 14:56:45:891 2008 H3C POSA/7/PACKET: Received 584 bytes from fcm terminal 2.
ID(0x60) DST(0xaaaa) SRC(0x1111)
Total length: 584 Offset: 0, partial data as follows:
0x000: 60 aa aa 11 11 30 6d 1b 5b 30 30 6d 61 6e 61 63
0x010: 6f 6e 64 61 2d 6b 73 2e 63 66 67 1b 5b 30 30 6d.
// POS access received a 584-byte packet.
```
The output description tables in this document only contain fields and messages that require an explanation.

**PPP debugging commands**

**debugging ppp**

Use `debugging ppp` to enable PPP debugging.

Use `undo debugging ppp` to disable PPP debugging.

**Syntax**

```plaintext

undo debugging ppp { all [ interface interface-type interface-number ] | bcp | cbcp packet | interface interface-type interface-number ] | ccp | chap | compression iphc { rtp | tcp } | core event [ interface interface-type interface-number ] | ip packet { interface interface-type interface-number ] | ipcp | ipv6 | ipv6cp | ipx | ipxcp | lcp | lqc | mp | mpls-multicast packet { interface interface-type interface-number ] | mpls-unicast packet { interface interface-type interface-number ] | mplscep | osi-npdu | osicp | pap | scp | vjcomp { all | error | event | packet | state } [ interface interface-type interface-number ] } 
```

**Default**

All types of PPP debugging are disabled.

**Views**

User view

**Default command level**

1. Monitor level

**Parameters**

- `all`: All types of PPP debugging.
- `bcp`: PPP Bridging Control Protocol (BCP) debugging.
- `cbcp`: PPP Callback Control Protocol (CBCP) debugging.
- `ccp`: PPP Compression Control Protocol (CCCP) debugging.
- `chap`: PPP CHAP debugging.
- `compression`: PPP IP header compression (TCP and RTP) debugging.
**core:** PPP kernel event debugging.

**ip:** IP packet debugging.

**ipcp:** IP Control Protocol (IPCP) debugging.

**ipv6:** IPv6 packet debugging.

**ipv6cp:** IPv6 Control Protocol (IPv6CP) debugging.

**ipx:** IPX debugging.

**ipxcp:** IPX Control Protocol (IPXCP) debugging.

**lcp:** PPP Link Control Protocol (CCP) debugging.

**lqc:** PPP link quality control (LQC) debugging.

**mp:** MP debugging.

**mpls-multicast:** MPLS multicast debugging.

**mpls-unicast:** MPLS unicast debugging.

**mplscp:** MPLS Control Protocol (MPLSCP) debugging.

**osi-npdu:** OSI NPDU debugging.

**osicp:** OSI Control Protocol (OSICP) debugging.

**pap:** PAP debugging.

**scp:** Stac LZS debugging.

**vjcomp:** VJ TCP/IP header compression debugging.

**error:** PPP error debugging.

**event:** PPP event debugging.

**packet:** PPP packet debugging.

**state:** PPP state debugging.

`interface-type interface-number`: Specifies an interface by its type and number.

### Usage guidelines

Table 2 describes the output fields and messages for the **debugging ppp event** command.

#### Table 146 Output from the debugging ppp event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>A PPP event occurred.</td>
</tr>
<tr>
<td>state</td>
<td>The state of a PPP state machine.</td>
</tr>
<tr>
<td>Up</td>
<td>The lower layer went up.</td>
</tr>
<tr>
<td>Down</td>
<td>The lower layer went down.</td>
</tr>
<tr>
<td>Open</td>
<td>The link was administratively opened.</td>
</tr>
<tr>
<td>Close</td>
<td>The link was administratively closed.</td>
</tr>
<tr>
<td>Timeout(T0+,T0-)</td>
<td>A timeout event occurred. T0+ indicates that the restart counter is greater than 0, so retransmission is required. T0- indicates that the restart counter is less than 0, so retransmission is not needed.</td>
</tr>
</tbody>
</table>
### Table 2: Output from the debugging ppp ipcp command

<table>
<thead>
<tr>
<th>Field value</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>IP-Compression-Protocol</td>
<td>The adopted IP compression protocol.</td>
</tr>
<tr>
<td>3</td>
<td>IP Address</td>
<td>IP address negotiation.</td>
</tr>
<tr>
<td>129</td>
<td>Primary DNS Server Address</td>
<td>PPP requested or allocated the primary DNS server.</td>
</tr>
<tr>
<td>130</td>
<td>Primary NBNS Server Address</td>
<td>PPP requested or allocated the primary NBNS server.</td>
</tr>
<tr>
<td>131</td>
<td>Secondary DNS Server Address</td>
<td>PPP requested or allocated a secondary DNS server.</td>
</tr>
<tr>
<td>132</td>
<td>Secondary NBNS Server Address</td>
<td>PPP requested or allocated a secondary NBNS server.</td>
</tr>
</tbody>
</table>

### Examples

Enable LCP debugging. Output similar to the following example is generated when you enable PPP on the two Serial interfaces connecting two devices and PPP negotiation starts between them.

```bash
<Sysname> debugging ppp lcp all
*0.784906 Sysname PPP/8/debug2:
  PPP Event:
    Serial2/0 LCP Open  Event
    state initial
```

// On interface Serial 2/0, the LCP state machine was opened and was in the initial state.
*0.784906 Sysname PPP/8/debug2:
PPP State Change:
    Serial2/0 LCP : initial --> starting
// LCP moved from the initial state to the starting state.
*0.784906 Sysname PPP/8/debug2:
PPP Event:
    Serial2/0 LCP Lower Up Event
state starting
// A lower layer up event was reported for LCP. The LCP state machine was in the starting state.
*0.784906 Sysname PPP/8/debug2:
PPP State Change:
    Serial2/0 LCP : starting --> reqsent
// LCP moved from the starting state to the reqsent state.
*0.784906 Sysname PPP/8/debug2:
PPP Packet:
    Serial2/0 Output LCP(c021) Pkt, Len 35
    State reqsent, code ConfReq(01), id 2a, len 31
    MRU(1), len 4, val 05dc
    AuthProto(3), len 4, PAP c023
    MagicNumber(5), len 6, val 31180c00
    MPRU(11), len 4, val 05dc
    Discrl(13), len 9, val 01fa4d432c8451
// Interface Serial 2/0 sent a 35-byte LCP packet. The type of the packet is Configure-Request, its ID is 2a, and its length is 31 bytes with the header removed. The LCP state machine transited to the request sent state as a result.

**PPPoE debugging commands**

The PPPoE client module name is identified as “PPPOEC” in debugging messages.

The PPPoE server module name is identified as “PPPOES” in debugging messages.

depending pppoe-client

Use **debugging pppoe-client** to enable PPPoE client debugging.

Use **undo debugging pppoe-client** to disable PPPoE client debugging.

**Syntax**

```
debugging pppoe-client { all | data | error | event | packet | verbose } [ interface interface-type interface-number ]
undo debugging pppoe-client { all | data | error | event | packet | verbose } [ interface interface-type interface-number ]
```

**Default**

All types of PPPoE client debugging are disabled.

**Views**

User view
Default command level

1. Monitor level

Parameters

all: All types of PPPoE client debugging.
data: Debugging for data information of the PPPoE client at the session stage.
error: Error debugging for the PPPoE client.
event: Event debugging for the PPPoE client.
packet: Debugging for data information of the PPPoE client at the discovery stage.
verbose: Detailed information debugging for the PPPoE client.

interface-type interface-number: Specifies an interface by its type and number.

Usage guidelines

Table 3 describes the output fields and messages for the debugging pppoe-client command.

Table 148 Output from the debugging pppoe-client command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>Incoming packets at the PPPoE session stage.</td>
</tr>
<tr>
<td>OUT</td>
<td>Outgoing packets at the PPPoE session stage.</td>
</tr>
<tr>
<td>Session Number</td>
<td>Session number.</td>
</tr>
<tr>
<td>Len</td>
<td>Packet length.</td>
</tr>
</tbody>
</table>

Table 4 describes the output fields and messages for the debugging pppoe-client packet command.

Table 149 Output from the debugging pppoe-client packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>Incoming packets at the PPPoE discovery stage.</td>
</tr>
<tr>
<td>OUT</td>
<td>Outgoing packets at the PPPoE discovery stage.</td>
</tr>
<tr>
<td>PADI</td>
<td>The PPPoE active discovery initiation (PADI) packet.</td>
</tr>
<tr>
<td>PADO</td>
<td>The PPPoE active discovery offer (PADO) packet.</td>
</tr>
<tr>
<td>PADR</td>
<td>The PPPoE active discovery request (PADR) packet.</td>
</tr>
<tr>
<td>PADS</td>
<td>The PPPoE active discovery session-conformation (PADS) packet.</td>
</tr>
<tr>
<td>PADT</td>
<td>The PPPoE active discovery terminate (PADT) packet.</td>
</tr>
<tr>
<td>Len</td>
<td>Packet length.</td>
</tr>
</tbody>
</table>

Table 5 describes the output fields and messages for the debugging pppoe-client verbose command.

Table 150 Output from the debugging pppoe-client verbose command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>Incoming packets at the PPPoE discovery stage.</td>
</tr>
<tr>
<td>OUT</td>
<td>Outgoing packets at the PPPoE discovery stage.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Len</td>
<td>Packet length.</td>
</tr>
<tr>
<td>Dest MAC Addr</td>
<td>Destination MAC address.</td>
</tr>
<tr>
<td>Src MAC Addr</td>
<td>Source MAC address.</td>
</tr>
<tr>
<td>Ver</td>
<td>Version.</td>
</tr>
<tr>
<td>Type</td>
<td>Type.</td>
</tr>
<tr>
<td>Code</td>
<td>Packet Type.</td>
</tr>
<tr>
<td>Session ID</td>
<td>Session number.</td>
</tr>
<tr>
<td>Tag Service-Name</td>
<td>Internet service provider.</td>
</tr>
<tr>
<td>Tag Host-Uniq</td>
<td>A particular unique request of a host.</td>
</tr>
<tr>
<td>Tag AC-Name</td>
<td>The field used for preventing denial of service attacks.</td>
</tr>
</tbody>
</table>

Examples

# Enable all types of PPPoE client debugging. Output similar to the following example is generated when you enable PPPoE on Ethernet interfaces connecting two devices and PPPoE negotiation starts between them.

*Aug 21 11:05:25:200 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client 0
UT Discovery packet (PADI), Len = 30

*Aug 21 11:05:25:201 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client 0
UT, Len = 30

    ff ff ff ff ff ff 00 0f e2 00 00 03 88 63 11 09
    00 00 00 0a 01 01 00 00 01 03 00 02 00 01
    >>
    Dest MAC Addr: ffff.ffff.ffff,    Src MAC Addr: 000f.e200.0003
    Discovery Stage,   Ver=1,   Type=1,   Code=PADI,   Session ID=0
    Tag Service-Name:
    Tag Host-Uniq:
        00 01 | ..

// Interface Ethernet 1/1 sent a discovery-stage packet.
*Aug 21 11:05:25:202 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client 0
IN, Len = 60

    00 0f e2 00 00 03 00 0f e2 29 ad f4 88 63 11 07
    00 00 00 21 01 01 00 00 01 03 00 02 00 01 01 02
    00 13 51 75 69 64 77 61 79 30 30 30 56 65 39 61
    00 39 61 64 66 34 00 00 00 00 00 00 00 00
    >>
    Dest MAC Addr: 000f.e200.0003,    Src MAC Addr: 000f.e229.adf4
    Discovery Stage,   Ver=1,   Type=1,   Code=PADO,   Session ID=0
    Tag Service-Name:
    Tag Host-Uniq:
        00 01 | ..
    Tag AC-Name:
        51 75 69 64 77 61 79 30 30 30 66 65 32 32 39 61 | Sysname000fe229a

255
*Aug 21 11:05:25:202 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client IN Discovery packet (PADO), Len = 60

// Interface Ethernet 1/1 received a discovery-stage packet.
*Aug 21 11:05:25:203 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client OUT Discovery packet (PADR), Len = 53

00 0f e2 29 ad f4 00 0f e2 00 00 03 88 63 11 19
00 00 00 21 01 01 00 00 01 03 00 02 00 01 01 02
00 13 51 75 69 64 77 61 79 30 30 30 66 65 32 32
39 61 64 66 34
>>
Dest MAC Addr: 000f.e229.adf4, Src MAC Addr: 000f.e200.0003
Discovery Stage, Ver=1, Type=1, Code=PADR, Session ID=0
Tag Service-Name:
  00 01 | ..
Tag Host-Uniq:
  51 75 69 64 77 61 79 30 30 30 66 65 32 32 39 61 | Sysname000fe229a
  64 66 34 | df4

// Interface Ethernet 1/1 sent a discovery-stage packet.
*Aug 21 11:05:25:208 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client IN, Len = 60
00 0f e2 00 00 03 00 0f e2 29 ad f4 88 63 11 65
00 01 00 21 01 01 00 00 01 03 00 02 00 01 01 02
00 13 51 75 69 64 77 61 79 30 30 30 66 65 32 32
39 61 64 66 34 00 00 00 00 00 00 00
>>
Dest MAC Addr: 000f.e200.0003, Src MAC Addr: 000f.e229.adf4
Discovery Stage, Ver=1, Type=1, Code=PADS, Session ID=1
Tag Service-Name:
  00 01 | ..
Tag Host-Uniq:
  51 75 69 64 77 61 79 30 30 30 66 65 32 32 39 61 | Sysname000fe229a
  64 66 34 | df4

*Aug 21 11:05:25:209 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client IN Discovery packet (PADS), Len = 60, Session ID = 1

// Interface Ethernet 1/1 received a discovery-stage packet.
*Aug 21 11:05:25:750 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client OUT session1 data, Dialer1:0, Len = 16

*Aug 21 11:05:26:400 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: PPPoE Client O
UT, Len = 36
00 0f e2 29 ad f4 00 0f e2 00 00 03 88 64 11 00
00 01 00 10 c0 21 01 00 00 0e 01 04 05 d4 05 06
3c f3 37 79
>>
Dest MAC Addr: 000f.e229.adf4, Src MAC Addr: 000f.e200.0003
Session Stage, Ver=1, Type=1, Session ID=1

*Aug 21 11:05:26:550 2007 Sysname PPPOEC/7/debugging:Ethernet1/1: Dialer1:0 was bound to Dialer1 for PPPoE Session

// Interface Ethernet 1/1 sent a session-stage packet.

debugging pppoe-server

Use `debugging pppoe-server` to enable PPPoE server debugging.

Use `undo debugging pppoe-server` to disable PPPoE server debugging.

Syntax

```plaintext
debugging pppoe-server { all | data | error | event | packet } [ interface interface-type interface-number ]
undo debugging pppoe-server { all | data | error | event | packet } [ interface interface-type interface-number ]
```

Default

All types of PPPoE server debugging is disabled.

Views

- **User view**
  - Default command level
    - 1. Monitor level

Parameters

- `all`: All types of PPPoE server debugging.
- `data`: Debugging for data information of the PPPoE server at the session stage.
- `error`: Error debugging for the PPPoE server.
- `event`: Event debugging for the PPPoE server.
- `packet`: Debugging for data information of the PPPoE server at the discovery stage.
- `interface interface-type interface-number`: Specifies an interface by its type and number.

Usage guidelines

Table 6 describes the output fields and messages for the `debugging pppoe-server` command.

### Table 151 Output from the debugging pppoe-server command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>Incoming packets at the PPPoE session stage.</td>
</tr>
<tr>
<td>OUT</td>
<td>Outgoing packets at the PPPoE session stage.</td>
</tr>
</tbody>
</table>
Table 7 describes the output fields and messages for the **debugging pppoe-server event** command.

**Table 152 Output from the debugging pppoe-server event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session</td>
<td>Session ID.</td>
</tr>
<tr>
<td>IN</td>
<td>Incoming packets at the PPPoE discovery stage.</td>
</tr>
<tr>
<td>OUT</td>
<td>Outgoing packets at the PPPoE discovery stage.</td>
</tr>
<tr>
<td>PADI</td>
<td>The PPPoE active discovery initiation (PADI) packet.</td>
</tr>
<tr>
<td>PADO</td>
<td>The PPPoE active discovery offer (PADO) packet.</td>
</tr>
<tr>
<td>PADR</td>
<td>The PPPoE active discovery request (PADR) packet.</td>
</tr>
<tr>
<td>PADS</td>
<td>The PPPoE active discovery session-conformation (PADS) packet.</td>
</tr>
<tr>
<td>PADT</td>
<td>The PPPoE active discovery terminate (PADT) packet.</td>
</tr>
</tbody>
</table>

Table 8 describes the output fields and messages for the **debugging pppoe-server packet** command.

**Table 153 Output from the debugging pppoe-server packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>Incoming packets at the PPPoE discovery stage.</td>
</tr>
<tr>
<td>OUT</td>
<td>Outgoing packets at the PPPoE discovery stage.</td>
</tr>
<tr>
<td>len</td>
<td>Packet length.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable all types of PPPoE client debugging. Output similar to the following example is generated when you enable PPPoE on the two GigabitEthernet interfaces connecting two devices.

```
*Dec  8 07:59:44:405 2006 Sysname PPPOE/7/debug2:
2006-12-8 7:59:44.404: PPPoE :GigabitEthernet1/1 IN discovery packet, len 60
 ff ff ff ff ff 00 0f e2 00 00 03 88 63 11 09
 00 00 00 0a 01 01 00 00 01 03 00 02 00 01 00 01
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*Dec  8 07:59:44:405 2006 Sysname PPPOE/7/debug2:
2006-12-8 7:59:44.404: PPPoE Event: GigabitEthernet1/1, IN PADI packet
// Interface GigabitEthernet 1/1 received a discovery-stage packet.
*Dec  8 07:59:44:405 2006 Sysname PPPOE/7/debug2:
2006-12-8 7:59:44.404: PPPoE :GigabitEthernet1/1 OUT discovery packet, len 53
 00 0f e2 00 00 03 00 0f e2 29 ad f4 88 63 11 07
 00 00 00 21 01 01 00 00 01 03 00 02 00 01 01 02
 00 13 51 75 69 64 77 61 79 30 30 30 66 65 32 32
 39 61 64 66 34
*Dec  8 07:59:44:405 2006 Sysname PPPOE/7/debug2:
2006-12-8 7:59:44.404: PPPoE Event: GigabitEthernet1/1, OUT PADO packet
```
// Interface GigabitEthernet 1/1 sent a discovery-stage packet.
*Dec 8 07:59:44:407 2006 Sysname PPPOE/7/debug2:
  2006-12-8 7:59:44.405: PPPoE :GigabitEthernet1/1 IN discovery packet, len 60
  00 0f e2 29 ad f4 00 0f e2 00 00 03 88 63 11 19
  00 00 00 21 01 01 00 00 01 03 00 02 00 01 01 02
  00 13 51 75 69 64 77 61 79 30 30 30 66 65 32 39
  61 64 66 34 00 13 51 75 69 64 77

*Dec 8 07:59:44:407 2006 Sysname PPPOE/7/debug2:
  2006-12-8 7:59:44.405: PPPoE Event: GigabitEthernet1/1, IN PADR packet

// Interface GigabitEthernet 1/1 received a discovery-stage packet.
*Dec 8 07:59:44:407 2006 Sysname PPPOE/7/debug2:
  2006-12-8 7:59:44.406: PPPoE :GigabitEthernet1/1 OUT discovery packet, len 53
  00 0f e2 00 00 03 00 0f e2 29 ad f4 88 63 11 65
  00 01 00 21 01 01 00 00 01 03 00 02 00 01 01 02
  00 13 51 75 69 64 77 61 79 30 30 30 66 65 32 32
  39 61 64 66 34

*Dec 8 07:59:44:407 2006 Sysname PPPOE/7/debug2:
  2006-12-8 7:59:44.406: PPPoE Event: GigabitEthernet1/1, session 1(Virtual-Temp
  late1:0), OUT PADS packet

// Interface GigabitEthernet 1/1 sent a discovery-stage packet.
// Now, the PPPoE server moves from the discovery stage to the session stage.
*Dec 8 07:59:44:898 2006 Sysname PPPOE/7/debug2:
  2006-12-8 7:59:44.410: PPPoE: GigabitEthernet1/1 IN session data for Virtual-T
  emplate1:0 (session 1), len 36
  00 0f e2 29 ad f4 00 0f e2 00 00 03 88 64 11 00
  00 01 00 10 c0 21 01 00 00 0e 01 04 05 d4 05 06
  3c d0 b6 97

// Interface GigabitEthernet 1/1 received a session-stage packet.
PTP debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging ptp

Syntax

depth 

undo debugging ptp 

Default

PTP debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all: All types of debugging for PTP.
error: PTP error debugging.
event: PTP event debugging.
frame: PTP message debugging.
receive: Debugging for received PTP messages.
send: Debugging for sent PTP messages.
verbose: Detailed PTP message debugging.
fsm: PTP state machine debugging.
timer: Debugging for PTP timers.

Usage guidelines

Use debugging ptp to enable PTP debugging. Use undo debugging ptp to disable PTP debugging.

Table 1 describes output fields and messages for the debugging ptp error command.

Table 154 Output from the debugging ptp error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to initialize PTP task information error, and failed to malloc memory.</td>
<td>Failed to initialize PTP task information and failed to allocate memory.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the debugging ptp fsm command.
### Table 155 Output from the debugging ptp fsm command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entered PRS</td>
<td>PTP entered role selection state.</td>
</tr>
<tr>
<td>Entered SEND_PDELAY_REQ state in PDelayReq state machine.</td>
<td>The Pdelay_Req state machine is in Pdelay request sending state.</td>
</tr>
<tr>
<td>Entered WAITING_FOR_PDELAY_RESP state in PDelayReq state machine.</td>
<td>The Pdelay_Req state machine is in waiting for the Pdelay response state.</td>
</tr>
<tr>
<td>Entered SENT_PDELAY_RESP state in PDelayResp state machine.</td>
<td>The Pdelay_Req state machine is in Pdelay response sending state.</td>
</tr>
<tr>
<td>Entered WAITING_FOR_PDELAY_INTERVAL_TIMER state in PDelayReq state machine.</td>
<td>The Pdelay_Req state machine is in waiting for the Pdelay interval timer state.</td>
</tr>
</tbody>
</table>

### Examples

#### # Enable PTP error debugging.
```
<Sysname> debugging ptp error
*Apr 2 15:39:33:219 2010 Sysname PTP1/7/ERROR : PTP is not enabled on Eth1/1.
```

#### # Enable PTP event debugging.
```
<Sysname> debugging ptp event
*Apr 2 15:18:34:375 2010 Sysname PTP1/7/EVENT : Created PTP task successfully.
*Apr 2 15:18:34:375 2010 Sysname PTP1/7/EVENT : Created PTP timer queue successfully.
*Apr 2 15:18:34:391 2010 Sysname PTP1/7/EVENT : Created PTP event queue successfully.
*Apr 2 15:18:34:391 2010 Sysname PTP1/7/EVENT : Created PTP timer successfully.
```

#### # Enable PTP message debugging.
```
<Sysname> debugging ptp frame receive verbose
*Apr 2 11:10:03:281 2010 Sysname PTP1/7/FRAME : -Slot=1; Ethernet1/1 received Announce packet [length 76].
```

#### # Enable PTP state machine debugging.
```
<Sysname> debugging ptp fsm
```

// PTP entered the Pdelay response sending state in PDelay_Resp state machine.
```
```

// PTP entered the state of waiting for the Pdelay interval timer in the Pdelay_Req state machine.
```
```

#### # Enable PTP timer debugging.
```
<Sysname> debugging ptp timer
```

// PTP entered the Pdelay response sending state in PDelay_Resp state machine.
QoS debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

NOTE:
Support for the following commands depends on the device model.

debugging dar

Use `debugging dar` to enable DAR debugging.

Use `undo debugging dar` to disable DAR debugging.

Syntax

```
debugging dar { all | error | event | packet }
undo debugging dar { all | error | event | packet }
```

Default

DAR debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `all`: All types of DAR debugging.
- `error`: Debugging for DAR errors.
- `event`: Debugging for DAR events.
- `packets`: Debugging for DAR packets.

Usage guidelines

Table 1 describes output fields and messages for the `debugging dar error` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to copy the data of the packet</td>
<td>DAR failed to copy the data of the packet during protocol recognition.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the `debugging dar event` command.
Table 157 Output from the debugging dar event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to write queue</td>
<td>DAR failed to write information queue during traffic accounting.</td>
</tr>
<tr>
<td>Failed to trigger an event</td>
<td>DAR failed to trigger an event after writing information queue successfully.</td>
</tr>
<tr>
<td>Failed to remove timer</td>
<td>DAR failed to remove the timer for traffic accounting.</td>
</tr>
<tr>
<td>Session count has reached the upper limit</td>
<td>The number of sessions monitored by DAR reached the configured upper limit.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the debugging dar packet command.

Table 158 Output from the debugging dar packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The head of the packet does not match</td>
<td>The packet header did not match.</td>
</tr>
</tbody>
</table>

Examples

# Enable debugging for DAR events. When traffic accounting is enabled on Ethernet 1/0, output similar to the following example is generated:

```bash
<Sysname> debugging dar event
*Apr 26 05:35:24:873 2007 Sysname DAR/7/DAR_Debug_Event: Ethernet1/0
  Succeeded in starting protocol recognition!
```

// The protocol recognition function was enabled successfully on Ethernet 1/0.

# Enable debugging for DAR packets. Traffic accounting is enabled on Ethernet 1/0. Output similar to the following example is generated when the IP address of Ethernet 1/0 is pinged from the device connected to Ethernet 1/0.

```bash
<Sysname> debugging dar packet
*Apr 26 05:40:28:630 2007 Sysname DAR/7/DAR_Debug_Packet: icmp
  Succeeded in protocol recognition: icmp.
```

// ICMP packet was recognized successfully.

debugging p2p

Use debugging p2p to enable P2P debugging.

Use undo debugging p2p to disable P2P debugging.

Syntax

```
debugging p2p { all | error | event }
undo debugging p2p { all | error | event }
```

Default

P2P debugging is disabled.

Views

User view
Default command level

2: System level

Parameters

- **all**: All types of P2P debugging.
- **error**: P2P error debugging.
- **event**: P2P event debugging.

Usage guidelines

Table 4 describes output fields and messages for the `debugging p2p error` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to parse the protocol string</td>
<td>P2P failed to resolve the protocol string.</td>
</tr>
<tr>
<td>Failed to get the protocol Chinese description from signature file</td>
<td>P2P failed to obtain the protocol description in Chinese from the signature file.</td>
</tr>
<tr>
<td>Failed to get the protocol English description from signature file</td>
<td>P2P failed to obtain the protocol description in English from the signature file.</td>
</tr>
<tr>
<td>Failed to synchronize the number of recognized packets</td>
<td>P2P failed to synchronize the number of recognized packets to the data plane.</td>
</tr>
</tbody>
</table>
| Failed to get the parameter `parameter-name` from signature file | P2P failed to obtain a parameter from the signature file. Parameters include:  
  - Starting position of the signature code.  
  - Offset of the signature code  
  - Signature code length. |
| Failed to synchronize the signature | P2P failed to synchronize the signature code to the data plane. |

Examples

# Enable P2P error debugging. The signature file `meta.mtd` became invalid because it was modified by the user without permission. When the modified signature file `meta.mtd` is loaded, output similar to the following example is generated:

```
<Sysname> debugging p2p error
```
Invalid signature file content.

// Contents of the signature file were invalid.

# Enable P2P event debugging when no protocol group exists on the device. Output similar to the following example is generated when a user performs these tasks:

- Creates protocol group 64.
- Adds protocol 1, protocol 2, protocol 3, and protocol 4 to protocol group 64.

debugging qos all

Use debugging qos all to enable all types of QoS debugging.

Use undo debugging qos all to disable all types of QoS debugging.

Syntax

d debuggqggs qos all

undo debugging qos all
Default

All types of QoS debugging are disabled.

Views

User view

Default command level

1: Monitor level

debugging qos car

Use `debugging qos car` to enable CAR debugging.

Use `undo debugging qos car` to disable CAR debugging.

Syntax

`debugging qos car`

`undo debugging qos car`

Default

CAR debugging is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

Table 5 describes output fields and messages for the `debugging qos car` command.

Table 160 Output from the debugging qos car command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropped a packet for CAR action</td>
<td>A packet was dropped by a CAR action configured on the interface.</td>
</tr>
</tbody>
</table>

Examples

# Enable CAR debugging on Router A. Output similar to the following example is generated when the rate of traffic from Router C to Router A exceeds 8 kbps under the following conditions:

- The traffic from Router C travels by way of Router A to Router B.
- A CAR policy is configured on the egress port of Router A to limit the outgoing traffic rate to 8 kbps.

```
<RouterA> debugging qos car
*Apr 26 05:19:13:589 2007 RouterA QOS/7/QOS_Debug:
QOS_CAR: Dropped a packet for CAR action
```

// A packet was dropped by CAR.

debugging qos cbq

Use `debugging qos cbq` to enable CBQ debugging.
Use `undo debugging qos cbq` to disable CBQ debugging.

**Syntax**

```
debugging qos cbq { af | be | ef }
undo debugging qos cbq { af | be | ef }
```

**Default**

CBQ debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- `af`: AF queue debugging.
- `be`: BE queue debugging.
- `ef`: EF queue debugging.

**Usage guidelines**

Table 6 describes output fields and messages for the `debugging qos cbq af` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropped a packet of class id <code>classID</code> from the AF queue for QoS cache was exhausted, and the current length is <code>queueLength</code></td>
<td>A packet matching class <code>classID</code> was dropped by the AF queue because the global QoS cache was exhausted. The current AF queue length is <code>queueLength</code>.</td>
</tr>
<tr>
<td>Dropped a packet of class id <code>classID</code> from the AF queue for zero bandwidth, and the current length is <code>queueLength</code></td>
<td>A packet matching class <code>classID</code> was dropped by the AF queue because of zero bandwidth. The current AF queue length is <code>queueLength</code>.</td>
</tr>
<tr>
<td>Dropped a packet of class id <code>classID</code> from the AF queue for tail-drop, and the current length is <code>queueLength</code></td>
<td>A packet matching class <code>classID</code> was dropped by the AF queue in the way of tail-drop. The current AF queue length is <code>queueLength</code>.</td>
</tr>
<tr>
<td>Dropped a packet of class id <code>classID</code> from the AF queue for WRED, and the current length is <code>queueLength</code></td>
<td>A packet matching class <code>classID</code> was dropped by the AF queue in the way of WRED. The current AF queue length is <code>queueLength</code>.</td>
</tr>
<tr>
<td>Dropped a packet of class id <code>classID</code> from the AF queue for illegal drop-type, and the current length is <code>queueLength</code></td>
<td>A packet matching class <code>classID</code> was dropped by the AF queue because of its illegal drop type. The current AF queue length is <code>queueLength</code>.</td>
</tr>
<tr>
<td>Dropped a packet of class id <code>classID</code> from the AF queue for insufficient packet wrapper memory, and the current length is <code>queueLength</code></td>
<td>A packet matching class id <code>classID</code> was dropped by the AF queue because of insufficient packet wrapper memory. The current length is <code>queueLength</code>.</td>
</tr>
</tbody>
</table>

**Examples**

The output in the following example was created when the following conditions exist:

- Packets sourced from Router C pass through Router A and Router B to reach Router D.

---

267
The packets from Router C are classified into three classes according to their DSCP values. A QoS policy is applied to the packets in the outbound direction of the egress port of Router A:

- Applying assured forwarding (AF) policy for the packets with the DSCP value AF11 or AF21, setting the minimum to bandwidth percentage to 5%.
- Applying expedited forwarding (EF) policy to the packets with the DSCP value EF, setting the maximum bandwidth percentage to 30%.

# Enable CBQ AF debugging and CBQ EF debugging on Router A. When the traffic of an AF class exceeds the minimum bandwidth and the traffic of the EF class exceeds the maximum bandwidth, the following information is generated:

```
<RouterA> debugging qos cbq af
<RouterA> debugging qos cbq ef
*Apr 26 05:19:13:589 2007 RouterA qos/7/qos_Debug:
  QOS_CBQEF: Dropped a packet of class id 4 from the EF queue for tokenbucket, and the current length is 1
// Packets matching class ID 4 were dropped by the EF queue due to token bucket restrictions, and the current EF queue length was 1.
*Apr 26 05:22:38:885 2007 RouterA QOS/7/QOS_Debug:
  QOS_CBQAF: Dropped a packet of class id 2 from the AF queue for tail-drop, and the current length is 64
// Packets matching class 2 were dropped by the AF queue in the way of tail-drop, and the current AF queue length is 64.
```

### debugging qos error

Use **debugging qos error** to enable QoS error debugging.

Use **undo debugging qos error** to disable QoS error debugging.

**Syntax**

```
depending qos error
undo debugging qos error
```

**Default**

QoS error debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Usage guidelines**

Table 7 describes output fields and messages for the **debugging qos error** command.

**Table 162 Output from the debugging qos error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting input packet link-layer-head length failed</td>
<td>QoS failed to obtain the link layer header length of incoming packets.</td>
</tr>
<tr>
<td>Creating MQC flow-statistic timer failed</td>
<td>QoS failed to create an MQC flow-statistics timer.</td>
</tr>
</tbody>
</table>
debugging qos event

Use **debugging qos event** to enable QoS event debugging.

Use **undo debugging qos event** to disable QoS event debugging.

**Syntax**

```
debbuging qos event
undo debugging qos event
```

**Default**

QoS event debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Usage guidelines**

Table 8 describes output fields and messages for the **debugging qos event** command.

### Table 163 Output from the **debugging qos event** command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting queue statistic error</td>
<td>QoS failed to obtain queue statistics.</td>
</tr>
<tr>
<td>Global fragment pre-drop count error</td>
<td>A global count error occurred on an interface enabled with fragment pre-drop.</td>
</tr>
<tr>
<td>Cutting packet header error on QoS input process</td>
<td>QoS failed to remove a header of an incoming packet.</td>
</tr>
<tr>
<td>Pasting packet header error on QoS input process</td>
<td>QoS failed to add a header to an incoming packet.</td>
</tr>
</tbody>
</table>

**Examples**

```
# Enable QoS event debugging. When CAR is configured on an interface, output similar to the following example is generated:
<Sysname> debugging qos event
```

269
debugging qos gts

Use `debugging qos gts` to enable GTS debugging.

Use `undo debugging qos gts` to disable GTS debugging.

**Syntax**

```plaintext
debugging qos gts
undo debugging qos gts
```

**Default**

GTS debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Examples**

```
# Enable GTS debugging on Router A. Output similar to the following example is generated when the rate of traffic from Router C to Router A exceeds 8 kbps under the following conditions:

- The traffic from Router C travels by way of Router A to Router B.
- A QoS policy that uses GTS is applied to the egress port of Router A to limit the outgoing traffic rate to 8 kbps.

<RouterA> debugging qos gts
```

**debugging qos match**

Use `debugging qos match` to enable debugging for packet matching in QoS.

Use `undo debugging qos match` to disable debugging for packet matching in QoS.

**Syntax**

```plaintext
debugging qos match
undo debugging qos match
```
Debugging for packet matching in QoS is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

Table 9 describes output fields and messages for the `debugging qos match` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match match-type were ignored for match-type did not match the packet.</td>
<td>The match criterion based on match-type was ignored because the match-type did not match the packet type.</td>
</tr>
</tbody>
</table>

Examples

# Enable debugging for packet matching in QoS on Router A. Output similar to the following example is generated when an IP packet arrives at Router A under the following conditions:
- The traffic from Router C travels by way of Router A to Router B.
- A QoS policy is applied to the egress port (GE port) of Router A to match outgoing packets by FR-DE.

<RouterA> debugging qos match

*Dec 17 14:05:31:150 2007 RouterA QOS/7/QOS_Debug:
QOS_Match: Match FR-DE were ignored for match-type did not match the packet.
// The FR-DE match criterion was ignored because the match-type FR-DE did not match the packet type.

debugging qos remark

Use `debugging qos remark` to enable debugging for packet re-marking in QoS.

Use `undo debugging qos remark` to disable debugging for packet re-marking in QoS.

Syntax

```
debugging qos remark
undo debugging qos remark
```

Default

Debugging for packet re-marking in QoS is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

Table 10 describes output fields and messages for the `debugging qos remark` command.
Table 165 Output from the debugging qos remark command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to remark packet, for remark remark-type did not match the packet type.</td>
<td>QoS failed to re-mark the packets because remark-type did not match the packet type.</td>
</tr>
</tbody>
</table>

Examples

# Enable debugging for packet remarking in QoS on Router A. Output similar to the following example is generated when an MPLS packet arrives at Router A under the following conditions:

- The traffic from Router C (PE) travels by way of Router A (P device) to Router B (PE).
- A QoS policy is applied to the egress port (GE port) of Router A to re-mark the IP precedence of outgoing packets as 5.

<RouterA> debugging qos remark
*Dec 17 14:05:01:720 2007 RouterA QOS/7/QOS_Debug:
QOS_Remark: Failed to remark packet, for remark ip-precedence did not match the packet type.

// QoS failed to set the IP precedence to 5 for the packets because the IP precedence type did not match the packet type.
Voice RADIUS debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

**debugging voice radius**

Use `debugging voice radius` to enable voice RADIUS debugging.

Use `undo debugging voice radius` to disable voice RADIUS debugging.

**Syntax**

```
debugging voice radius { all | error | event | fsm | timer | trace }
undo debugging voice radius { all | error | event | fsm | timer | trace }
```

**Default**

Voice RADIUS debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **all**: Specifies all types of debugging for voice RADIUS.
- **error**: Specifies error debugging.
- **event**: Specifies event debugging.
- **fsm**: Specifies finite state machine debugging.
- **timer**: Specifies timer debugging.
- **trace**: Specifies debugging for messages between the voice RADIUS module and the AAA module.

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging voice radius error` command.

### Table 166 Output from the debugging voice radius error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arg Error</td>
<td>The input parameter is incorrect.</td>
</tr>
<tr>
<td>Play Voice Error</td>
<td>Playing prompt tones failed.</td>
</tr>
<tr>
<td>Msg Not Support In This State</td>
<td>A message that is not supported in the current state was received.</td>
</tr>
<tr>
<td>Create Timer Fail</td>
<td>Creating a timer failed.</td>
</tr>
<tr>
<td>Start Authen Error</td>
<td>Originating an authentication request failed.</td>
</tr>
</tbody>
</table>
Table 2 describes output fields and messages for the **debugging voice radius event** command.

Table 167 **Output from the debugging voice radius event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disconnect Media Error</td>
<td>Disconnecting the media failed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VORDS -&gt; CMC</td>
<td>Commands from the voice RADIUS module to the CMC.</td>
</tr>
<tr>
<td>VORDS -&gt; CMC</td>
<td>The voice RADIUS module invoked a CMC interface to establish a unidirectional connection.</td>
</tr>
<tr>
<td>VORDS -&gt; CMC Connect Media</td>
<td>The voice RADIUS module invoked a CMC interface to connect the media.</td>
</tr>
<tr>
<td>VORDS -&gt; CMC Create New Call</td>
<td>The voice RADIUS module invoked a CMC interface to establish a new call.</td>
</tr>
<tr>
<td>VORDS -&gt; CMC Start Trigger</td>
<td>The voice RADIUS module invoked a CMC interface to trigger a service.</td>
</tr>
<tr>
<td>VORDS -&gt; CMC End Trigger</td>
<td>The voice RADIUS module invoked a CMC interface to stop triggering a service.</td>
</tr>
<tr>
<td>VORDS -&gt; CMC Terminate Call</td>
<td>The voice RADIUS module invoked a CMC interface to end a call.</td>
</tr>
<tr>
<td>VORDS -&gt; CMC Get History Info</td>
<td>The voice RADIUS module invoked a CMC interface to obtain call history information.</td>
</tr>
<tr>
<td>VORDS Voice -&gt; IVR: Play</td>
<td>The voice RADIUS module invoked an IVR interface to play prompts.</td>
</tr>
<tr>
<td>VORDS Voice -&gt; IVR Stop</td>
<td>The voice RADIUS module invoked an IVR interface to stop playing voice prompts.</td>
</tr>
<tr>
<td>VORDS -&gt; DRV Play SigTone</td>
<td>The voice RADIUS invoked a drive interface to play signal tones.</td>
</tr>
<tr>
<td>VORDS -&gt; DRV Stop SigTone</td>
<td>The voice RADIUS module invoked a drive interface to stop playing signal tones.</td>
</tr>
<tr>
<td>VORDS -&gt; DRV Start Dtmf Detect</td>
<td>The voice RADIUS module invoked a drive interface to start DTMF detection.</td>
</tr>
<tr>
<td>VORDS -&gt; DRV Stop Dtmf Detect</td>
<td>The voice RADIUS module invoked a drive interface to stop DTMF detection.</td>
</tr>
<tr>
<td>Dial Flow Type: Did</td>
<td>The dialing process is a two-stage dialing process.</td>
</tr>
</tbody>
</table>

Dial Flow Type: Did

The dialing process is a two-stage dialing process.
- **Field** | **Description**
- Dial Flow Type: Card | The dialing process is a card number/password process (two-stage dialing process).
- Dial Flow Type: CallerNumber | The dialing process is a caller number process (two-stage dialing process).
- Dial Flow Type: VoiceCaller | The dialing process is a caller number process with IVR (two-stage dialing process).

Table 3 describes output fields and messages for the **debugging voice radius fsm** command.

**Table 168 Output from the debugging voice radius fsm command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[RDS_STATE_INIT] -&gt; [RDS_STATE_SELECT_LANG]</td>
</tr>
<tr>
<td>RDS_STATE_INIT</td>
<td>Initial state.</td>
</tr>
<tr>
<td>RDS_STATE_SELECT_LANG</td>
<td>A language was selected for prompt tones.</td>
</tr>
<tr>
<td>RDS_STATE_CARD_RCV</td>
<td>Receiving digits of a card number.</td>
</tr>
<tr>
<td>RDS_STATE_PWD_RCV</td>
<td>Receiving digits of the password.</td>
</tr>
<tr>
<td>RDS_STATE_CALLED_RCV</td>
<td>Receiving digits of the called number.</td>
</tr>
<tr>
<td>RDS_STATE_TALKING</td>
<td>Conversation state.</td>
</tr>
<tr>
<td>RDS_STATE_RELEASE</td>
<td>Call release state.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the **debugging voice radius timer** command.

**Table 169 Output from the debugging voice radius timer command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create Timer: RDS_AUTHEN_TIMER ID: 3 Length: 4ms</td>
</tr>
<tr>
<td>1</td>
<td>Delete Timer: RDS_AUTHEN_TIMER ID: 3</td>
</tr>
<tr>
<td>RDS_INIT_AAA_TIMER</td>
<td>Timer that specifies a wait for an AAA_OK message in the initial state.</td>
</tr>
<tr>
<td>RDS_FIRSTNUM_TIMER</td>
<td>Timer that specifies a wait for the first digit of a called number or card number.</td>
</tr>
<tr>
<td>RDS_INTERNUM_TIMER</td>
<td>Timer that specifies a wait for a digit other than the first one of a called number or card number.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RDS_AUTHEN_TIMER</td>
<td>Timer that specifies a wait for an Authentication_OK message.</td>
</tr>
<tr>
<td>RDS_AUTHOR TIMER</td>
<td>Timer that specifies a wait for an Authorization_OK message.</td>
</tr>
<tr>
<td>RDS_ACCT TIMER</td>
<td>Timer that specifies a wait for an Accounting_OK message.</td>
</tr>
<tr>
<td>RDS_MDACONN TIMER</td>
<td>Timer that specifies a wait for a Media_Connect_OK message.</td>
</tr>
<tr>
<td>RDS_TALK TIMER</td>
<td>Timer that specifies the call duration.</td>
</tr>
<tr>
<td>RDS_LASTMINUTE TIMER</td>
<td>Timer that specifies the last minute of a call.</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the **debugging voice radius trace** command.

**Table 170 Output from the debugging voice radius trace command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Msg: ACCT_START</td>
<td>Accounting_Start request from the voice RADIUS module to the AAA module.</td>
</tr>
<tr>
<td>Msg: ACCT_STOP:</td>
<td>Accounting_Stop request from the voice RADIUS module to the AAA module.</td>
</tr>
<tr>
<td>Msg: AUTHORIZATION</td>
<td>Authorization request from the voice RADIUS module to the AAA module.</td>
</tr>
<tr>
<td>Msg: AUTHEN</td>
<td>Authentication request from the voice RADIUS module to the AAA module.</td>
</tr>
<tr>
<td>Msg: LEAVING</td>
<td>Leaving request from the voice RADIUS module to the AAA module.</td>
</tr>
<tr>
<td>Msg: ACCT_FAIL</td>
<td>Accounting_Fail from the AAA module to the voice RADIUS module.</td>
</tr>
<tr>
<td>Msg: ACCT_OK</td>
<td>Accounting_OK from the AAA module to the voice RADIUS module.</td>
</tr>
<tr>
<td>Msg: AUTHEN_OK</td>
<td>Authentication_OK from the AAA module to the voice RADIUS module.</td>
</tr>
<tr>
<td>Msg: AUTHEN_FAIL</td>
<td>Authentication_Fail from the AAA module to the voice RADIUS module.</td>
</tr>
<tr>
<td>Msg: AUTHOR_OK</td>
<td>Authorization_OK from the AAA module to the voice RADIUS module.</td>
</tr>
<tr>
<td>Msg: AUTHOR_FAIL</td>
<td>Authorization_Fail from the AAA module to the voice RADIUS module.</td>
</tr>
<tr>
<td>Msg: CUT</td>
<td>Cut command from the AAA module to the voice RADIUS module.</td>
</tr>
</tbody>
</table>
Examples

The output in the following examples was created when the following conditions exist:

- A two-stage dialing process is configured for voice RADIUS.
- The access number is 201.
- The card number/password authentication, authorization, and accounting functions are enabled.

# Enable voice RADIUS event debugging.

```bash
<Sysname> debugging voice radius event
Enable VORDS event debugging functions
```

* Nov 17 17:40:30:86 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: VORDS ---> CMC Start Trigger. CMC ID: [0] VORDS ID: [0]
  // The voice RADIUS module invoked the CMC interface to trigger a service.

* Nov 17 17:40:30:87 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_SETUP Calling: 200, Called: 201 CIT Index: 0
  // The voice RADIUS module received a CMC_SETUP message from the CMC. The calling number is
  // 200, and the access number is 201.

* Nov 17 17:40:30:87 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: AAA SetUserID [0] OK, VORDS [0], VORDS AAAID: [0] [-1]
  // The AAA module invoked the voice RADIUS interface to set the ID of the module that interacts with
  // the AAA module.

* Nov 17 17:40:30:88 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: [0] [RDS_STATE_INIT] VORDS ---> CMC UniConnect. CMC ID: [0]
  // The voice RADIUS module invoked the CMC interface to establish a unidirectional connection.

* Nov 17 17:40:30:89 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: ACCP_CHANNEL_READY
  // An ACCP_CHANNEL_READY message was received from the CMC.

* Nov 17 17:40:30:89 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: VORDS ---> IVR: Play Voice. VORDS ID: [0] State: [RDS_STATE_SELECT_LANG] IfIndex: [0x2c0030] Tone ID: [1100]
  // The voice RADIUS invoked the IVR interface to play the prompt for a language.

* Nov 17 17:40:30:90 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: DTMF Detect. VORDS ID: [0] IfIndex: [0x2c0030]
  // DTMF detection was started for voice interface 0x2c0030.

* Nov 17 17:40:35:65 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: DTMFChar: [1]
  // The user pressed the digit 1, and the voice RADIUS module received a key message and played the
  // prompt in Chinese.

* Nov 17 17:40:35:66 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  // The user pressed any key to stop playing the prompt.
// A prompt was played for a card number.

*Nov 17 17:40:38:68 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: Receive IVR Message. VORDS ID: [0] Argument: [1]
  // A prompt response message was received from the interactive voice response (IVR). The result of
  prompt play was 1, which means that the prompt was played successfully.

*Nov 17 17:40:38:695 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_INFORMATION DTMFChar: [4]
  // The digit 4 pressed was received by the user.

*Nov 17 17:40:39:115 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_INFORMATION DTMFChar: [5]

*Nov 17 17:40:39:565 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_INFORMATION DTMFChar: [0]

*Nov 17 17:40:40:15 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_INFORMATION DTMFChar: [1]

*Nov 17 17:40:40:405 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_INFORMATION DTMFChar: [1]

*Nov 17 17:40:40:885 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_INFORMATION DTMFChar: [0]

*Nov 17 17:40:42:295 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_INFORMATION DTMFChar: [#]
  // The information above is the card number collection process.

*Nov 17 17:40:42:295 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: VORDS ---> IVR: Play Voice. VORDS ID: [0] State: [RDS_STATE_PWD_RCV]
  IfIndex: [0x2c0030] Tone ID: [1400]

// A prompt was played for a password.

*Nov 17 17:40:45:200 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: Receive IVR Message. VORDS ID: [0] Argument: [1]
  // A prompt response message was received from the IVR. The result of prompt play was 1, which means
  that the prompt was played successfully. If the result had been 2, the prompt was not played successfully.

*Nov 17 17:40:48:235 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT: CMC [0] ---> VORDS [0]
  Msg: CMC_INFORMATION DTMFChar.
The information above is the password collection process.

The AAA module invoked the RADIUS interface to set an AAA ID.

The prompt was played to announce the balance in the card.

The prompt was played for a called number.

The user dialed the first digit 4 of the called number.
// The information above is the called number collection process. After the user pressed the dial terminator #, the called number collection process ended.

*Nov 17 17:41:05:187 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  VORDS ---> DRV  Stop DTMF Detect.  VORDS ID: [0]  IfIndex:[0x2c0030]

// DTMF detection stopped.
*Nov 17 17:41:05:187 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  VORDS ---> IVR  Stop Voice.  VORDS ID: [0]  State: [RDS_STATE_TALKING]  IfIndex: [0x2c0030]

// Going into conversation and stopping the playing of any prompt.
*Nov 17 17:41:05:188 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  VORDS ---> CMC  Create New Call.  VORDS ID: [0]  CMC ID: [0]  Called: [400]  CIT: [1]

// The CMC interface was invoked to create a connection for a new outgoing call.
*Nov 17 17:41:05:338 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  CMC [1] ---> VORDS [0]
  Msg: CMC_ALERTING  InBandInfo: Invalid

// An Alerting message was received, which means that the in-band information is invalid and the voice RADIUS module was required to play ringback tones.
*Nov 17 17:41:05:338 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  VORDS ---> DRV  Play SigTone.  VORDS ID: [0]  State: [5]  Type: [RingBack Tone]  IfIndex: [0x2c0030]

// Ringback tones were played.
*Nov 17 17:41:11:129 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  CMC [1] ---> VORDS [0]
  Msg: CMC_CONNECT

// A CMC_CONNECT (callee offhook) message was received from the CMC.
*Nov 17 17:41:11:129 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  VORDS ---> DRV  Stop SigTone.  VORDS ID: [0]  State: [5]  IfIndex:[0x2c0030]
  SigTone Type: [RingBack Tone]

// The voice RADIUS module ordered the driver to stop playing ringback tones, and the conversation began.
*Nov 17 17:41:11:130 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  VORDS ---> CMC  Connect Media.  VORDS ID: [0]  CMC ID: [0][1].

// The media channel was connected.
*Nov 17 17:41:17:490 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  CMC [1] ---> VORDS [0]
  Msg: ACCP_RELEASE

*Nov 17 17:41:17:491 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  CMC [1] ---> VORDS [0]
  Msg: CMC_RELEASE  Release Cause: Normal clearing!

// After a period of time, the called user hung up, and the voice RADIUS module received the CMC_Release message.
*Nov 17 17:41:17:492 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  VORDS ---> IVR:  Play Voice.  VORDS ID: [0]  State: [RDS_STATE_TALKING]  IfIndex: [0x2c0030]  Tone ID: [2084]

280
The prompt was played to notify the voice RADIUS module that the called user had hung up.

Nov 17 17:41:20:161 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  Receive IVR Message.  VORDS ID: [0]  Argument: [1]

DTMF detection was enabled to start collecting the digits of a called number.

Nov 17 17:41:20:161 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:  VORDS ---> DRV  Start DTMF Detect.  VORDS ID: [0]  IfIndex: [0x2c0030]

The prompt was played for the called number.

Nov 17 17:41:23:311 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:   Receive IVR Message.  VORDS ID: [0]  Argument: [1]

Nov 17 17:41:25:122 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:    CMC [0] ---> VORDS [0]
      MSG: ACCP_RELEASE

The calling user hung up while dialing the called number.

Nov 17 17:41:25:123 2006 Sysname VORDS/7/VOICE:
  VORDS_EVENT:    CMC [0] ---> VORDS [0]
      MSG: CMC_RELEASE  Release Cause: Normal clearing!

Enable debugging for voice RADIUS finite state machines.

# Enable debugging for voice RADIUS finite state machines.
<Sysname> debugging voice radius fsm
 Enable VORDS fsm debugging functions

The state of voice RADIUS module 1 changed from RDS_STATE_INIT to RDS_STATE_INIT.

Nov 17 17:44:17:963 2006 Sysname VORDS/7/VOICE:
  VORDS_FSM:  [1] RDS_STATE_INIT ---> RDS_STATE_INIT

The state of voice RADIUS module 1 changed from RDS_STATE_INIT to RDS_STATE_SELECT_LANG.

Nov 17 17:44:17:964 2006 Sysname VORDS/7/VOICE:
  VORDS_FSM:  [1] RDS_STATE_INIT ---> RDS_STATE_SELECT_LANG

The state of voice RADIUS module 1 changed from RDS_STATE_SELECT_LANG to RDS_STATE_CARD_RCV.

Nov 17 17:44:23:602 2006 Sysname VORDS/7/VOICE:
  VORDS_FSM:  [1] RDS_STATE_SELECT_LANG ---> RDS_STATE_CARD_RCV

The state of voice RADIUS module 1 changed from RDS_STATE_CARD_RCV to RDS_STATE_PWD_RCV.

Nov 17 17:44:29:242 2006 Sysname VORDS/7/VOICE:
  VORDS_FSM:  [1] RDS_STATE_CARD_RCV ---> RDS_STATE_PWD_RCV
Nov 17 17:44:39:919 2006 Sysname VORDS/7/VOICE:

VORDS_FSM: [1] RDS_STATE_PWD_RCV ---> RDS_STATE_CALLED_RCV

// The state of voice RADIUS module changed from RDS_STATE_PWD_RCV to RDS_STATE_CALLED_RCV.

Nov 17 17:44:50:402 2006 Sysname VORDS/7/VOICE:

VORDS_FSM: [1] RDS_STATE_CALLED_RCV ---> RDS_STATE_TALKING

// The state of voice RADIUS module 1 changed from RDS_STATE_CALLED_RCV to RDS_STATE_TALKING.

Nov 17 17:45:05:635 2006 Sysname VORDS/7/VOICE:

VORDS_FSM: [1] RDS_STATE_TALKING ---> RDS_STATE_CALLED_RCV

// The state of voice RADIUS module 1 changed from RDS_STATE_TALKING to RDS_STATE_CALLED_RCV.

Nov 17 17:45:09:99 2006 Sysname VORDS/7/VOICE:

VORDS_FSM: [1] RDS_STATE_CALLED_RCV ---> RDS_STATE_RELEASE

// The state of voice RADIUS module 1 changed from RDS_STATE_CALLED_RCV to RDS_STATE_RELEASE.

# Enable voice RADIUS timer debugging.
<Sysname> debugging voice radius timer
Enable VORDS timer debugging functions
<Sysname>

Nov 17 17:45:43:292 2006 Sysname VORDS/7/VOICE:


// A timer was created to specify the wait time for a Media_Connect_OK message as 15 seconds.

Nov 17 17:43:45:293 2006 Sysname VORDS/7/VOICE:


// The timer for waiting for a Media_Connect_OK message was deleted.

Nov 17 17:44:51:167 2006 Sysname VORDS/7/VOICE:

VORDS_TIMER: [2] [RDS_STATE_CARD_RCV]    Create Timer: RDS_FIRSTNUM_TIMER ID: 300 Length: 10000 ms

// A timer was created, specifying a wait time for the first digit of a card number.

Nov 17 17:44:51:501 2006 Sysname VORDS/7/VOICE:

VORDS_TIMER: [2] [RDS_STATE_CARD_RCV]    Delete Timer: RDS_FIRSTNUM_TIMER ID: 300 Length: 10000 ms

// The RDS_FIRSTNUM_TIMER was deleted after the first digit of the card number was received.

Nov 17 17:45:51:201 2006 Sysname VORDS/7/VOICE:

VORDS_TIMER: [2] [RDS_STATE_CARD_RCV]    Create Timer: RDS_INTERNUM_TIMER ID: 303 Length: 10000 ms

// A timer was created, specifying a wait time for a digit other than the first one of a card number.

Nov 17 17:45:51:501 2006 Sysname VORDS/7/VOICE:

VORDS_TIMER: [2] [RDS_STATE_CARD_RCV]    Delete Timer: RDS_INTERNUM_TIMER ID: 303

// The RDS_INTERNUM_TIMER was deleted each time the user dialed a digit.
The information above is the card number collection process. When a digit of the card number was received, an RDS_INTERNUM_TIMER was deleted. The process stopped when all digits of the card number were received.

A timer was created, specifying a wait time for the first digit of a password.
The information above is the password collection process. When a digit of the password was received, an RDS_INTERNUM_TIMER was deleted. The process stopped when all digits of the password were received.

The password collection process ended. A card number/password authentication request was originated to the AAA module. An RDS_AUTHEN_TIMER was created.

The RDS_AUTHEN_TIMER was deleted after an Authentication_OK message was received.

A timer was created, specifying a wait time for the first digit of a called number.
The information above is the called number collection process. After the user pressed the dial terminator #, all digits of the called number were received, and the RDS_FIRSTNUM_TIMER and RDS_INTERNUM_TIMER were deleted.

The digits of the called number were received. An authorization request was originated to the AAA module. The RDS_AUTHOR_TIMER with a time length of 10 seconds was created.

An Authorization_OK message was received from the AAA module. The RDS_AUTHOR_TIMER was deleted.

The RDS_TALK_TIMER with a time length of 8 seconds was created.

The RDS_TALK_TIMER was deleted.

A timer was created, specifying a wait time for a VoIP_Accounting_Stop_ACK message.

The timer for waiting for a VoIP_Accounting_Stop_ACK message was deleted.

A timer was created, specifying a wait time for the first digit of a called number.
The timer for waiting for the first digit of a called number was deleted.

A timer was created, specifying a wait time for a PSTN_Accounting_Stop_ACK message.

The timer for waiting for a PSTN_Accounting_Stop_ACK message was deleted.

Enable debugging for messages between the voice RADIUS module and the AAA module.

The voice RADIUS module sent an Accounting_Start message to the AAA module. The accounting type is 1, which means that accounting was performed on the originating PSTN side.

A PSTN_Accounting_OK message was received from the AAA module.

The password was received, and a card number/password authentication request was originated to the AAA module.

An Authentication_OK message was received from the AAA module. The balance on the card is RMB71.

An Authorization message was sent to the AAA module for the called number.

An Authorization_OK message was received from the AAA module. The call duration was 60 seconds.
Nov 17 17:47:31:739 2006 Sysname VORDS/7/VOICE:
   Msg: ACCT_START   Calling Num: 200   Called Num: 400
      Acct Type: 2   Access Num: 201
// A VoIP_Accounting_Start message was sent to the AAA module.

Nov 17 17:47:31:790 2006 Sysname VORDS/7/VOICE:
   Msg: ACCT_OK   Acct Type: 2
// A VoIP_Accounting_OK message was received. Accounting succeeded.

Nov 17 17:47:43:142 2006 Sysname VORDS/7/VOICE:
   Msg: ACCT_STOP   Calling Num: 200   Called Num: 400
      Acct Type: 2   Access Num: 201   Stop-only: [No]
      Connection Time: 13   Session Time: 5
// A VoIP_Accounting_Stop message was sent to the AAA module. The connection time is 13 seconds, and the call duration is 5 seconds.

Nov 17 17:47:43:202 2006 Sysname VORDS/7/VOICE:
   Msg: ACCT_OK   Acct Type: 2
// An Accounting_OK message was received from the AAA module.

Nov 17 17:47:47:706 2006 Sysname VORDS/7/VOICE:
   Msg: ACCT_STOP   Calling Num: 200   Called Num: 201
      Acct Type: 1   Access Num: 201   Stop-only: [No]
      Connection Time: 52   Session Time: 9
// A PSTN_Accounting_Stop message was sent to the AAA module.

Nov 17 17:47:47:783 2006 Sysname VORDS/7/VOICE:
   Msg: ACCT_OK   Acct Type: 1
// An Accounting_OK message was received from the AAA module.

Nov 17 17:47:47:784 2006 Sysname VORDS/7/VOICE:
   Msg: LEAVING   Calling Num: 200   Called Num: 201
// A Leaving message was sent to the AAA module for the incoming call.

Nov 17 17:47:47:784 2006 Sysname VORDS/7/VOICE:
   Msg: LEAVING   Calling Num: 200   Called Num: 201
// A LEAVING message was sent to the AAA module for the outgoing call.
RIP debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging rip

Use `debugging rip` to enable RIP debugging.
Use `undo debugging rip` to disable RIP debugging.

**Syntax**

```
debugging rip process-id
undo debugging rip process-id
```

**Default**

RIP debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

`process-id`: Specifies a RIP process by its ID in the range of 1 to 65535.

**Examples**

```
# Enable RIP debugging for RIP process 1.
<Sysname> debugging rip 1
```

debugging rip brief

Use `debugging rip brief` to enable brief RIP debugging.
Use `undo debugging rip brief` to disable brief RIP debugging.

**Syntax**

```
debugging rip process-id brief
undo debugging rip process-id brief
```

**Default**

RIP brief debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level
Parameters

process-id: Specifies a RIP process by its ID in the range of 1 to 65535.

Examples

# Enable brief RIP debugging on Router A. The output in this example was created when the following conditions exist:

- Router A is connected to Router B through Ethernet 1/1.
- RIP is enabled on Router A and Router B.

<Sysname> debugging rip 1 brief
*Nov 24 15:28:22:814 2006 Sysname RM/6/RMDEBUG: RIP 1 : Sending v2 response on Ethernet1/1 from 40.0.0.2

// RIP process 1 sent a response packet of version 2 on Ethernet 1/1 from 40.0.0.2.

*Nov 24 15:28:34:868 2006 Sysname RM/6/RMDEBUG: RIP 1 : Receiving v2 response on Ethernet1/1 from 40.0.0.1

// RIP process 1 received a response packet of version 2 on Ethernet 1/1 from 40.0.0.1.

debugging rip event

Use debugging rip event to enable RIP event debugging.

Use undo debugging rip event to disable RIP event debugging.

Syntax

debugging rip process-id event

undo debugging rip process-id event

Default

RIP event debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

process-id: Specifies a RIP process by its ID in the range of 1 to 65535.

Usage guidelines

Table 1 describes output fields and messages for the debugging rip event command.
### Table 171 Output from the debugging rip event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| RIP-BFD: Success to send msg. Message Type msg-type, Plt Type pkt-type, Src IP Address src-ip, Src IFIndex if-index, Nbr IP Address nbr-ip. | RIP sent a message to BFD successfully.  
- **msg-type**—BFD message type.  
- **pkt-type**—Packet type, which can be Echo or Control.  
- **src-ip**—Source IP address for session establishment.  
- **if-index**—Interface index maintained by route management.  
- **nbr-ip**—Neighbor IP address for session establishment. |
| RIP-BFD: Failed to send msg. Message Type msg-type, Plt Type pkt-type, Src IP Address src-ip, Src IFIndex if-index, Nbr IP Address nbr-ip. | RIP failed to send a message to BFD. |
| RIP-BFD: Receive BFD session down, Plt Type pkt-type, Src IP Address src-ip, Src IFIndex if-index, Dst IP Address nbr-ip. | RIP received a message indicating that the BFD session is down. |

### Examples

# Enable RIP event debugging on Router A. The output in this example was created when the following conditions exist:
- Router A is connected to Router B through Ethernet 1/1.
- RIP is enabled on Router A and Router B.

```bash
<Sysname> debugging rip 1 event
```

# On Router A, disable RIP process 1 on Ethernet 1/1 attached to the network 40.0.0.0.
```
[Sysname-rip-1] undo network 40.0.0.0
```

*Nov 24 15:33:24:194 2006 Sysname RM/6/RMDEBUG: RIP 1 : Removing Ethernet1/1 from Network List

// RIP process 1 was disabled on Ethernet 1/1.

# On Router A, enable RIP process 1 on Ethernet 1/1 attached to the network 40.0.0.0.
```
[Sysname-rip-1] network 40.0.0.0
```

*Nov 24 15:36:12:162 2006 Sysname RM/6/RMDEBUG: RIP 1 : Adding Ethernet1/1 to Network List

// RIP process 1 was enabled on Ethernet 1/1.

# Redistribute static routes into RIP process 1 on Router A.
```
[Sysname-rip-1] import-route static
```

*Nov 24 15:38:21:642 2006 Sysname RM/6/RMDEBUG: RIP 1 : Rebuilding of Database has started
*Nov 24 15:38:21:642 2006 Sysname RM/6/RMDEBUG: RIP 1 : Database has been rebuilt

// RIP process 1 had rebuilt the database.

# Enable RIP event debugging on Router A. The output in this example was created when the following conditions exist:
- Serial 2/0 of Router A is connected to Serial 2/0 of Router B.
- RIP process 1 is created on Router A and TRIP is enabled on Serial 2/0.
- RIP process is created on Router B and TRIP is enabled on Serial 2/0.
debugging rip packet

Use `debugging rip packet` to enable RIP packet debugging.
Use `undo debugging rip packet` to disable RIP packet debugging.

**Syntax**

```plaintext
debugging rip process-id packet [ interface interface-type interface-number ]
undo debugging rip process-id packet [ interface interface-type interface-number ]
```

**Default**

RIP packet debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- `process-id`: Specifies a RIP process by its ID in the range of 1 to 65535.
- `interface interface-type interface-number`: Specifies an interface by its type and number.

**Examples**

# Enable RIP packet debugging on Ethernet 1/1 of Router A. The output in this example was created when the following conditions exist:

- Router A is connected to Router B through Ethernet 1/1.
- RIP is enabled on Router A and Router B.

```
<Sysname> debugging rip 1 packet interface ethernet 1/1
```

```
*Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: RIP 1 : Sending response on interface Ethernet1/1 from 40.0.0.2 to 224.0.0.9

// RIP process 1 sent a response packet on Ethernet 1/1 from 40.0.0.2 to 224.0.0.9.
```

```
```

// The response packet version is 2 and the packet length is 84 bytes.

```
```

// The packet uses simple authentication, with password 1234.
Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 50.0.0.2/255.255.255.255, nethop 0.0.0.0, cost 1, tag 0

// The first route entry was output.

Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 50.0.0.0/255.0.0.0, nethop 0.0.0.0, cost 1, tag 0

// The second route entry was output.

Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 110.0.0.1/255.255.255.255, nethop 0.0.0.0, cost 1, tag 0

// The third route entry was output.

Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: RIP 1 : Receive response from 40.0.0.1 on Ethernet1/1

// RIP process 1 received a response packet from 40.0.0.1 on Ethernet 1/1.


// The version of the received response packet is 2 and the packet length is 84 bytes.

Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 50.0.0.1/255.255.255.255, nethop 0.0.0.0, cost 1, tag 0

// The first route entry was output.

Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 50.0.0.0/255.0.0.0, nethop 0.0.0.0, cost 1, tag 0

// The second route entry was output.

Nov 24 15:57:46:86 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 100.0.0.1/255.255.255.255, nethop 0.0.0.0, cost 1, tag 0

// The third route entry was output.

Nov 24 15:57:46:86 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 110.0.0.1/255.255.255.255, nethop 0.0.0.0, cost 16, tag 0

// The fourth route entry was output.


// Authentication failed. The received response packet was ignored.

# Clear the packet debugging information of the specified process, so no related debugging information is displayed.
<Sysname> reset rip 1 process
Warning : Reset RIP process? [Y/N]:y

# Enable packet debugging if you want to view packet debugging information on Router A after the reset operation.
<Sysname> debugging rip 1 packet

# Enable RIP packet debugging on Router A. The output in this example was created when the following conditions exist:
  • Serial 2/0 of Router A is connected to Serial 2/0 of Router B.
  • RIP is enabled on Router A and Router B.

<Sysname> debugging rip 1 packet

Nov 24 15:45:45:522 2006 Sysname RM/6/RMDEBUG: TRIP 1 : Sending request on interface Serial2/0 to 224.0.0.9

// TRIP process 1 sent a request packet on Serial2/0 to 224.0.0.9.
// The version of the sent request packet is 2 and the packet length is 8 bytes.
*Nov 24 15:45:45:564 2006 Sysname RM/6/RMDEBUG: TRIP 1 : Receive response on Serial2/0 from 50.0.0.2
// TRIP process 1 received a response packet from 50.0.0.2 on Serial 2/0.
*Nov 24 15:45:45:564 2006 Sysname RM/6/RMDEBUG:   Packet : vers 2, cmd response (FLUSH), length 8, sequence num 0
// The version of the received response packet (with FLUSH) is 2. The packet length is 8 bytes and the sequence number is 0.
*Nov 24 15:45:45:564 2006 Sysname RM/6/RMDEBUG: TRIP 1 : Sending acknowledgement on interface Serial2/0 to 50.0.0.2
// TRIP process 1 sent an acknowledgement packet on Serial 2/0 to 50.0.0.2.
*Nov 24 15:45:45:564 2006 Sysname RM/6/RMDEBUG:   Packet : vers 2, cmd acknowledgement (FLUSH), length 8, sequence num 0
// The version of the sent acknowledgement packet (with FLUSH) is 2. The packet length is 8 bytes and the sequence number is 0.

debugging rip receive

Use **debugging rip receive** to enable received RIP packet debugging.
Use **undo debugging rip receive** to disable received RIP packet debugging.

**Syntax**

```
debugging rip process-id receive [ interface interface-type interface-number ]
undo debugging rip process-id receive [ interface interface-type interface-number ]
```

**Default**

Received RIP packet debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **process-id**: Specifies a RIP process by its ID in the range of 1 to 65535.
- **interface interface-type interface-number**: Specifies an interface by its type and number.

**Usage guidelines**

Table 2 describes output fields and messages for the **debugging rip receive** command.

**Table 172 Output from the debugging rip receive command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication failure</td>
<td>Authentication failed because of an authentication type mismatch or a non-minus RFC 2453 sequence number.</td>
</tr>
</tbody>
</table>
### Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence: seq-number (number)</td>
<td>Sequence number of MD5 authentication (RFC 2453).</td>
</tr>
<tr>
<td>Sequence: seq-number</td>
<td>Sequence number of MD5 authentication (RFC 2082).</td>
</tr>
<tr>
<td>invalid authentication</td>
<td>The values of the 0xFFFF and 0x01 fields in the MD5 authentication (RFC 2082) packets are incorrect. Therefore, the authentication is invalid.</td>
</tr>
</tbody>
</table>

### Examples

# Enable RIP received packet debugging on Ethernet 1/1 of Router A. The output in this example was created when the following conditions exist:

- Router A is connected to Router B through Ethernet 1/1.
- RIP is enabled on Router A and Router B.

```bash
<Sysname> debugging rip 1 receive interface ethernet 1/1
```

// RIP process 1 received a response packet from 40.0.0.1 on Ethernet 1/1.

```bash
*Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: RIP 1: Receive response from 40.0.0.1 on Ethernet1/0
```

// The version of the received response packet is 2 and the packet length is 84 bytes.

```bash
```

// The first route entry was output.

```bash
*Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 50.0.0.1/255.255.255.255, nexthop 0.0.0.0, cost 1, tag 0
```

// The second route entry was output.

```bash
*Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 50.0.0.0/255.0.0.0, nexthop 0.0.0.0, cost 1, tag 0
```

// The third route entry was output.

```bash
*Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 100.0.0.1/255.255.255.255, nexthop 0.0.0.0, cost 16, tag 0
```

// The fourth route entry was output.

# Clear the packet debugging information of the specified process, so no related debugging information is displayed.

```bash
<Sysname> reset rip 1 process
Warning: Reset RIP process? [Y/N]: y
```

# Enable received packet debugging if you want to view packet debugging information on Router A after the reset operation.

```bash
<Sysname> debugging rip 1 receive
```

# Enable RIP received packet debugging on Router A. The output in this example was created when the following conditions exist:

- Serial 2/0 of Router A is connected to Serial 2/0 of Router B.
- RIP is enabled on Router A and Router B.

```bash
<Sysname> debugging rip 1 receive
```
debugging rip send

Use `debugging rip send` to enable sent RIP packet debugging.

Use `undo debugging rip send` to disable sent RIP packet debugging.

Syntax

```
debugging rip process-id send [ interface interface-type interface-number ]
undo debugging rip process-id send [ interface interface-type interface-number ]
```

Default

Sent RIP packet debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- **process-id**: Specifies a RIP process by its ID in the range of 1 to 65535.
- **interface interface-type interface-number**: Specifies an interface by its type and number.

Usage guidelines

Table 3 describes output fields and messages for the `debugging rip send` command.

### Table 173 Output from the `debugging rip send` command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence: seq-number (number)</td>
<td>Sequence number of MD5 authentication (RFC 2453).</td>
</tr>
<tr>
<td>Sequence: seq-number</td>
<td>Sequence number of MD5 authentication (RFC 2082).</td>
</tr>
<tr>
<td>invalid authentication</td>
<td>The values of the 0xFFFF and 0x01 fields in the MD5 authentication (RFC 2082) packets are incorrect. Therefore, the authentication is invalid.</td>
</tr>
</tbody>
</table>

Examples

# Enable RIP sent packet debugging on Ethernet 1/1 of Router A. The output in this example was created when the following conditions exist:

- Router A is connected to Router B through Ethernet 1/1.
- RIP is enabled on Router A and Router B.

```bash
<Sysname> debugging rip 1 send interface ethernet1/1
```
Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: RIP 1 : Sending response on interface Ethernet1/1 from 40.0.0.2 to 224.0.0.9

// RIP process 1 sent a response packet on Ethernet 1/1 from 40.0.0.2 to 224.0.0.9.

// The version of the response packet is 2 and the packet length is 84 bytes.

// The packet uses simple authentication, with the password 1234.
Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 50.0.0.2/255.255.255.255, nexthop 0.0.0.0, cost 1, tag 0

// The first route entry was output.
Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 50.0.0.0/255.0.0.0, nexthop 0.0.0.0, cost 1, tag 0

// The second route entry was output.
Nov 24 15:57:46:32 2006 Sysname RM/6/RMDEBUG: AFI 2, dest 110.0.0.1/255.255.255.255, nexthop 0.0.0.0, cost 1, tag 0

// The third route entry was output.

# Clear the packet debugging information of the specified process, so no related debugging information is displayed.
<Sysname> reset rip 1 process
Warning : Reset RIP process? [Y/N]:y

# Enable sent packet debugging if you want to view packet debugging information on Router A after the reset operation.
<Sysname> debugging rip 1 send

# Enable RIP sent packet debugging on Router A. The output in this example was created when the following conditions exist:
• Serial 2/0 of Router A is connected to Serial 2/0 of Router B.
• RIP is enabled on Router A and Router B.

<Sysname> debugging rip 1 send
Nov 24 15:45:45:522 2006 Sysname RM/6/RMDEBUG: TRIP 1 : Sending request on interface Serial2/0 to 224.0.0.9

// TRIP process 1 sent a request packet on Serial 2/0 to 224.0.0.9.
Nov 24 15:45:45:537 2006 Sysname RM/6/RMDEBUG: Packet : vers 2, cmd request, length 8

// The version of the sent request packet is 2 and the packet length is 8 bytes.
Nov 24 15:45:45:564 2006 Sysname RM/6/RMDEBUG: TRIP 1 : Sending acknowledgement on interface Serial2/0 to 50.0.0.2

// TRIP process 1 sent an acknowledgement packet on Serial 2/0 to 50.0.0.2.
Nov 24 15:45:45:564 2006 Sysname RM/6/RMDEBUG: Packet : vers 2, cmd acknowledgement (FLUSH), length 8, sequence num 0

// The version of the sent acknowledgement packet (with FLUSH) is 2. The packet length is 8 bytes and the sequence number is 0.
debugging rip timer

Use `debugging rip timer` to enable RIP timer debugging.

Use `undo debugging rip timer` to disable RIP timer debugging.

Syntax

`debugging rip process-id timer`

`undo debugging rip process-id timer`

Default

RIP timer debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

`process-id`: Specifies a RIP process by its ID in the range of 1 to 65535.

`interface interface-type interface-number`: Specifies an interface by its type and number.

Examples

# Enable timer debugging for RIP process 1.
<Sysname> debugging rip 1 timer


// The periodic update timer of RIP process 1 expired.
RIPng debugging commands

debugging ripng

Use **debugging ripng** to enable all RIPng debugging.
Use **undo debugging ripng** to disable all RIPng debugging.

**Syntax**

```
debugging ripng process-id
undo debugging ripng process-id
```

**Default**

No RIPng debugging is enabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

`process-id`: Specifies a RIPng process by its ID in the range of 1 to 65535.

debugging ripng brief

Use **debugging ripng brief** to enable RIPng brief debugging.
Use **undo debugging ripng brief** to disable RIPng brief debugging.

**Syntax**

```
debugging ripng process-id brief
undo debugging ripng process-id brief
```

**Default**

RIPng brief debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

`process-id`: Specifies a RIPng process by its ID in the range of 1 to 65535.

**Examples**

```
# Enable RIPng brief debugging on Router A. The output in this example was created when the following conditions exist:
```
• Ethernet 1/1 of Router A is connected to Ethernet 1/1 of Router B.
• RIPng is enabled on Ethernet 1/1 of Router A and on Ethernet 1/1 of Router B.

<Sysname> debugging ripng 1 brief
*Nov 22 21:17:37:662 2006 Sysname RM/6/RMDEBUG:RIPng 1 : Sending v1 response on Ethernet1/1 to FF02::9 with 2 RTEs

// RIPng process 1 sent a response packets of version 1 on Ethernet 1/1 to FF02::9 with two RTEs.
*Nov 22 21:17:40:390 2006 Sysname RM/6/RMDEBUG:RIPng 1 : Receiving v1 response on Ethernet1/1 from FE80::200:5EFF:FE71:A706 with 2 RTEs

// RIPng process 1 received a response packet of version 1 on Ethernet 1/1 from FE80::200:5EFF:FE71:A706 with two RTEs.

debugging ripng event

Use debugging ripng event to enable RIPng event debugging.
Use undo debugging ripng event to disable RIPng event debugging.

Syntax
debugging ripng process-id event
undo debugging ripng process-id event

Default
RIPng event debugging is disabled.

Views
User view

Default command level
1: Monitor level

Parameters
process-id: Specifies a RIPng process by its ID, in the range of 1 to 65535.

Examples
# Enable RIPng event debugging on Router A. The output in this example was created when the following conditions exist:
• RIPng process 1 is created on Router A.
• RIPng is enabled on Ethernet 1/1.
<Sysname> debugging ripng 1 event
[Sysname-Ethernet1/1] ripng 1 enable
*Nov 22 21:48:54:988 2006 Sysname RM/6/RMDEBUG:RIPng 1 : Adding Ethernet1/1 to Network List

// RIPng process 1 was enabled on Ethernet 1/1.

// RIPng process 1 sent a triggered update.
# Disable RIPng on Ethernet 1/1 of Router A.
[Sysname-Ethernet1/1] undo ripng enable
*Nov 22 21:50:46:270 2006 Sysname RM/6/RMDEBUG:RIPng 1 : Removing Ethernet1/1 from Network List
/ RIPng process 1 was disabled on Ethernet 1/1.
# Redistribute direct routes into RIPng process 1 on Router A.
[Sysname-ripng-1] import-route direct
*Nov 23 13:29:24:640 2006 Sysname RM/6/RMDEBUG:RIPng 1 : Rebuilding of Database has started
*Nov 23 13:29:24:640 2006 Sysname RM/6/RMDEBUG:RIPng 1 : Database has been rebuilt
// RIPng process 1 rebuilt the database.

debugging ripng packet

Use debugging ripng packet to enable RIPng packet debugging.
Use undo debugging ripng packet to disable RIPng packet debugging.

Syntax
debugging ripng process-id packet [ interface interface-type interface-number ]
undo debugging ripng process-id packet [ interface interface-type interface-number ]

Default
RIPng packet debugging is disabled.

Views
User view

Default command level
1 : Monitor level

Parameters
process-id: Specifies a RIPng process by its ID, in the range of 1 to 65535.
interface interface-type interface-number: Specifies an interface by its type and number.

Examples
# Enable RIPng packet debugging on Router A. The output in this example was created when the
following conditions exist:
• Ethernet 1/1 of Router A is connected to Ethernet 1/1 of Router B.
• RIPng process 1 is created on Router A and RIPng is enabled on Ethernet 1/1.
• RIPng process is created on Router B and RIPng is enabled on Ethernet 1/1.
<Sysname> debugging ripng 1 packet
FE80::200:5EFF:FE71:A700 on Ethernet1/1
// RIPng process 1 received a response packet on Ethernet 1/1, whose IPv6 address is
FE80::200:5EFF:FE71:A700.
// The version of the received response packet is 1 and the packet length is 64 bytes.
*Nov 24 13:49:27:98 2006 Sysname RM/6/RMDEBUG: Dest 22::/64, cost 16, tag 0
// The first route entry was output.
*Nov 24 13:49:27:98 2006 Sysname RM/6/RMDEBUG: Dest 50::/64, cost 1, tag 0
// The second route entry was output.
*Nov 24 13:49:27:98 2006 Sysname RM/6/RMDEBUG: Dest 1001::1/128, cost 1, tag 0
debugging ripng receive

Use debugging ripng receive to enable RIPng received packet debugging.

Use undo debugging ripng receive to disable RIPng received packet debugging.

Syntax

dumping ripng process-id receive [ interface interface-type interface-number ]

undo dumping ripng process-id receive [ interface interface-type interface-number ]

Default

RIPng received packet debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

process-id: Specifies a RIPng process by its ID, in the range of 1 to 65535.

interface interface-type interface-number: Specifies an interface by its type and number.

Examples

# Enable RIPng received packet debugging on Router A. The output in this example was created when
the following conditions exist:
• Ethernet 1/1 of Router A is connected to Ethernet 1/1 of Router B.
• RIPng process 1 is created on Router A and RIPng is enabled on Ethernet 1/1.
• RIPng process 1 is created on Router B and RIPng is enabled on Ethernet 1/1.

<Sysname> debugging ripng 1 receive

*Nov 22 21:41:02:00 2006 Sysname RM/6/RMDEBUG:RIPng 1 : Receiving response message from
FE80::200:5EFF:FE71:A706 on Ethernet1/1

// RIPng process 1 received a response packet from FE80::200:5EFF:FE71:A706 on Ethernet 1/1.

*Nov 22 21:41:02:00 2006 Sysname RM/6/RMDEBUG: Packet : vers 1, cmd response, length 44

// The version of the received response packet is 1, and the packet length is 44 bytes.
debugging ripng send

Use **debugging ripng send** to enable RIPng sent packet debugging.

Use **undo debugging ripng send** to disable RIPng sent packet debugging.

**Syntax**

```
debugging ripng process-id send [ interface interface-type interface-number ]
undo debugging ripng process-id send [ interface interface-type interface-number ]
```

**Default**

RIPng sent packet debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **process-id**: Specifies a RIPng process by its ID, in the range of 1 to 65535.
- **interface interface-type interface-number**: Specifies an interface by its type and number.

**Examples**

```
# Enable RIPng sent packet debugging on Router A. The output in this example was created when the following conditions exist:
  • Ethernet 1/1 of Router A is connected to Ethernet 1/1 of Router B.
  • RIPng process 1 is created on Router A and RIPng is enabled on Ethernet 1/1.
  • RIPng process is created on Router B and RIPng is enabled on Ethernet 1/1.

<Sysname> debugging ripng 1 send
*Nov 22 21:35:29:86 2006 Sysname RM/6/RMDEBUG: RIPng 1 : Sending response message on Ethernet1/1 to FF02::9
  // RIPng process 1 sent a RIPng response packet on Ethernet 1/1 to FF02::9.
  // The response packet version is 1, and the packet length is 44 bytes.
*Nov 22 21:35:29:86 2006 Sysname RM/6/RMDEBUG: Dest 22::/64, cost 1, tag 0
  // The first route entry was output.
*Nov 22 21:35:29:86 2006 Sysname RM/6/RMDEBUG: Dest 33::/64, cost 1, tag 0
  // The second route entry was output.
```

debugging ripng timer

Use **debugging ripng timer** to enable RIPng timer debugging.

Use **undo debugging ripng timer** to disable RIPng timer debugging.
**Syntax**

```
debugging ripng process-id timer
undo debugging ripng process-id timer
```

**Default**

RIPng timer debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

`process-id`: Specifies a RIPng process by its ID in the range of 1 to 65535.

**Examples**

```
# Enable RIPng timer debugging on the device running RIPng.
<Sysname> debugging ripng 1 timer
// The periodic update timer of RIPng process 1 expired.
```
## RMON debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

### debugging rmon

Use `debugging rmon` to enable RMON debugging.

Use `undo debugging rmon` to disable RMON debugging.

#### Syntax

```
debugging rmon [ error | info | warning ]
undo debugging rmon [ error | info | warning ]
```

#### Views

User view

#### Default command level

1: Monitor level

#### Parameters

- `error`: Specifies RMON error debugging.
- `info`: Specifies RMON info debugging.
- `warning`: Specifies RMON warning debugging.

#### Usage guidelines

If you execute the command without specifying any keyword, the command enables all type of RMON debugging.

Table 1 describes output fields and messages for the `debugging rmon` command.

### Table 174 Output from the debugging rmon command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFO</td>
<td>Message severity level is informational.</td>
</tr>
<tr>
<td>WARNING</td>
<td>Message severity level is warning.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Message severity level is error.</td>
</tr>
<tr>
<td>reallocate memory for saving configuration</td>
<td>Memory was reallocated for saving RMON configuration.</td>
</tr>
<tr>
<td>failed to reallocate memory for saving configuration</td>
<td>Failed to reallocate memory for saving RMON configuration.</td>
</tr>
<tr>
<td>update current RMON configuration of <code>interface-name</code></td>
<td>RMON configuration was updated on an interface.</td>
</tr>
<tr>
<td>invalid index <code>index</code> in real time</td>
<td>Invalid RMON table entry index was found during real-time backup.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| invalid index index in real time on slave board | Invalid RMON table entry index was found on the standby MPU during real-time backup.  
  index—Index of an RMON table entry.                                                                                                                                 |
| etherStatsEntry index aged at time-hour:time-minute:time-second in real time on slave board | The Ethernet statistics entry aged out during real-time data backup on the standby MPU.                                                                 |
| historyControlEntry index aged at time-hour:time-minute:time-second in real time on slave board | The history control entry aged out on the standby MPU.                                                                                     |
| eventEntry index aged at time-hour:time-minute:time-second in real time on slave board | The event entry aged out during real-time data backup on the standby MPU.                                                                  |
| alarmEntry index aged at time-hour:time-minute:time-second in real time on slave board | The alarm entry aged out during real-time data backup on the standby MPU.                                                                  |
| prialarmEntry index aged at time-hour:time-minute:time-second in real time on slave board | The private alarm entry aged out during real-time data backup on the standby MPU.                                                           |
| the port-removed flag of etherStatsEntry index changed to false for interface-inserted event at time-hour:time-minute:time-second in real time on slave board | The port-removed flag of the Ethernet statistics entry on the standby MPU changed to false because an interface card was inserted during real-time data backup. |
| the port-removed flag of historyControlEntry index changed to false for interface-inserted event at time-hour:time-minute:time-second in real time on slave board | The port-removed flag of the history control entry on the standby MPU changed to false because an interface card was inserted during real-time data backup. |
| failed to recover data in real time on slave board | The standby MPU failed to restore data during real-time backup.                                                                               |
| etherStatsEntry index aged at time-hour:time-minute:time-second | Removal of an Ethernet interface can cause its Ethernet statistics entry to age out.                                                          |
| historyControlEntry index aged at time-hour:time-minute:time-second | A history control entry typically ages out because its bucket is full or its timer expires.                                                |
| synchronous RPC timeout in sampling statistics at interface-name for historyControlEntry index | RPC call timed out when RMON was sampling data on the Ethernet interface for the history control entry.                                           |
| failed to call RPC synchronously (errCode: error-code) in sampling statistics at interface-name for historyControlEntry index | A synchronous RPC call failed when RMON was sampling data on the Ethernet interface for the history control entry.                           |
| sample no statistics at interface-name for historyControlEntry index by synchronous RPC | RMON performed a synchronous RPC call to sample statistics for the history control entry on the Ethernet interface, but the call did not collect any statistics. |
| synchronous RPC timeout in sampling statistics at interface-name for etherStatsEntry index | Synchronous RPC call timed out when RMON sampled statistics on the Ethernet interface for the Ethernet statistics entry.                    |
| failed to call RPC synchronously (errCode: error-code) in sampling statistics at interface-name for etherStatsEntry index | A synchronous RPC call failed when RMON was sampling data on the Ethernet interface for the Ethernet statistics entry.                    |
RMON performed a synchronous RPC call to sample statistics for the Ethernet statistics entry on the Ethernet interface, but the call did not collect any statistics.

Removal of an interface card caused all its interfaces to be removed. As a result, the status of the Ethernet statistics entry for an interface on the card changed to underCreation. In this state, the entry does not take effect.

Removal of an interface card caused all its interfaces to be removed. As a result, the status of the history control entry for an interface on the card changed to underCreation. In this state, the entry does not take effect.

The port-removed flag of the Ethernet statistics entry changed to false because an interface card was inserted during real-time data backup.

The port-removed flag of the history control entry on the standby MPU changed to false because an interface card was inserted during real-time data backup.

An Ethernet statistics entry was deleted because its associated interface was deleted.

An interface delete event typically occurs when you replace an interface card with a different-model interface card.

A history control entry was deleted because its associated interface was deleted.

An interface delete event typically occurs when you replace an interface card with a different-model interface card.

The (ordinal-number)th sampling was performed for prialarmTable entry index.

Examples

# Enable displaying debugging information and system message monitoring on the current terminal.
<Sysname> terminal debugging
<Sysname> terminal monitor

# Enable RMON info debugging.
<Sysname> debugging rmon info

# Configure a valid event group and a valid alarm group.
<Sysname> system-view
[Sysname] rmon event 1 log owner test
[Sysname]
*Apr 27 10:11:54:56 2000 Sysname RMON/7/INFO:
  add eventEntry 1 through CLI or NMS

# Create an RMON alarm entry.
[Sysname] rmon alarm 1 1.3.6.1.2.1.6.3.0 5 absolute rising-threshold 100 1
falling-threshold 20 1

*Apr 27 10:12:24:208 2000 Sysname RMON/7/INFO:
    add alarmEntry 1 through CLI or NMS

*Apr 27 10:12:24:216 2000 Sysname RMON/7/INFO:
    the No. 1 sample for alarmEntry 1

// RMON created the RMON alarm entry and sampled the monitored object for the first time.

*Apr 27 10:12:28:403 2000 Sysname RMON/7/INFO:
    sample statistics for alarmEntry 1 at 10:12:28

*Apr 27 10:12:28:408 2000 Sysname RMON/7/INFO:
    the No. 2 sample for alarmEntry 1

*Apr 27 10:12:33:454 2000 Sysname RMON/7/INFO:
    sample statistics for alarmEntry 1 at 10:12:33

*Apr 27 10:12:33:459 2000 Sysname RMON/7/INFO:
    the No. 3 sample for alarmEntry 1

*Apr 27 10:12:38:504 2000 Sysname RMON/7/INFO:
    sample statistics for alarmEntry 1 at 10:12:38

*Apr 27 10:12:38:509 2000 Sysname RMON/7/INFO:
    the No. 4 sample for alarmEntry 1

// RMON periodically sampled the monitored object at the configured interval (5 seconds).

# Enable RMON error debugging.

<Sysname> debugging rmon error

<Sysname> system-view

# Configure an RMON private alarm entry with a nonexistent MIB variable.

[Sysname] rmon prialarm 1 100/(1.3.6.1.2.1.4.2.0-255) test 10 absolute rising-threshold 10 1
falling-threshold 20 1 entrytype forever

    Failure! Wrong private alarm variable formula.

[Sysname]

*Jul 26 16:43:46:703 2007 Sysname RMON/7/ERROR:
    failed to validate formula of prialarmEntry 1

// RMON failed to validate the formula of the monitored object in the newly created private alarm entry.
RPR debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging rpr error

Use debugging rpr error to enable RPR error debugging.

Use undo debugging rpr error to disable RPR error debugging.

Syntax

debugging rpr error [ interface interface-type interface-number ]
undo debugging rpr error [ interface interface-type interface-number ]

Default

RPR error debugging is disabled.

Views

User view

Default command level

2: System level

Parameters

interface interface-type interface-number: Specifies a port by its type and number.

Usage guidelines

Table 1 describes output fields and messages for the debugging rpr error command.

Table 175 Output from the debugging rpr error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPR_Error: On interface-type interface-number, String</td>
<td>System resource operation failed. The String field content failed to allocate memory.</td>
</tr>
</tbody>
</table>
| \text{String1} error packet is received. String2 | Error packets were received. The String1 field indicates the direction (east or west). The String2 field describes the detailed reason for errors:  
\begin{itemize} 
  \item \text{Invalid frame length. Type: packetType}—Frames of invalid length were received. packetType indicates the packet type, which can be TP, TC, ATD, echo-response, or echo-request. 
  \item \text{Invalid frame type}—Packets of invalid types were received. 
\end{itemize} |
## Field Description

Errors occurred to topology calculation.

The "String" field describes the detailed reason for errors:

- **MAC duplicate error**: ringlet ringlet_id hop hop_id duplicate with local station, which means the MAC address of the hop hop-id of ringlet ringlet-id overlaps with that of the local station. ringlet-id is 0 or 1, and hop-id ranges from 1 to 254.

- **MAC duplicate error**: ringlet ringlet_id1 hop hop_id1 duplicate with ringlet ringlet_id2 hop hop_id2, which means the MAC address of the hop hop-id1 of ringlet ringlet-id1 overlaps with that of the hop hop-id2 of ringlet ringlet-id2. Both ringlet-id1 and ringlet-id2 range from 0 to 1, and both hop-id1 and hop-id2 ranges from 1 to 254.

- **IP duplicate error**: ringlet ringlet_id hop hop_id duplicate with local station, which means the IP address of the hop hop-id of ringlet ringlet-id overlaps with that of the local station. ringlet-id is 0 or 1, and hop-id ranges from 1 to 254.

- **IP duplicate error**: ringlet ringlet_id1 hop hop_id1 duplicate with ringlet ringlet_id2 hop hop_id2, which means the IP address of the hop hop-id1 of ringlet ringlet-id1 overlaps with that of the hop hop-id2 of ringlet ringlet-id2. Both ringlet-id1 and ringlet-id2 range from 0 to 1, and both hop-id1 and hop-id2 ranges from 1 to 254.

### Examples

# Enable debugging for RPR errors on interface RPR1. When you configure the same IP address for the two stations that form a closed RPR ring, output similar to the following example is generated:

```plaintext
<Sysname> debugging rpr error interface rpr1
*Jun 5 10:06:39:104 2007 Sysname RPR/7/ERR_TOPO:Slot=3;
RPR_Error: On RPR1, topology error happened: Ip duplicate error: ringlet 0 hop 1 duplicate with local station
// Topology calculation errors occurred on interface RPR1: the IP address of the first hop of ringlet 0 overlapped with that of the local station.
```

### debugging rpr event

Use `debugging rpr event` to enable RPR event debugging.

Use `undo debugging rpr event` to disable RPR event debugging.

#### Syntax

```plaintext
default rpr event [ general | ringlet-selection ] [ interface interface-type interface-number ]
undo default rpr event [ general | ringlet-selection ] [ interface interface-type interface-number ]
```

#### Default

RPR event debugging is disabled.

#### Views

User view

#### Default command level

2: System level
Parameters

**general**: Specifies debugging for general RPR events, including protection state changes, miscabling changes, and port UP/DOWN events.

**ringlet-selection**: Specifies debugging for RPR ringlet selection events.

**interface interface-type interface-number**: Specifies a port by its type and number.

Usage guidelines

Table 2 describes output fields and messages for the **debugging rpr event** command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPR Event: On interface-type interface-number, String1</td>
<td>A ringlet selection table was updated. The String1 field specifies the updated ringlet selection table:</td>
</tr>
<tr>
<td></td>
<td>• default — Default ringlet selection table.</td>
</tr>
<tr>
<td></td>
<td>• shortest path — Shortest path ringlet selection table.</td>
</tr>
<tr>
<td></td>
<td>• overall — Overall ringlet selection table.</td>
</tr>
<tr>
<td></td>
<td>• VRRP — VRRP ringlet selection table.</td>
</tr>
<tr>
<td></td>
<td>• IPv6 — IPv6 ringlet selection table.</td>
</tr>
<tr>
<td></td>
<td>• VLANTunnel — VLAN tunnel ringlet selection table.</td>
</tr>
<tr>
<td></td>
<td>• mac-learning — MAC learning ringlet selection table.</td>
</tr>
<tr>
<td></td>
<td>The String2 field specifies the updated contents:</td>
</tr>
<tr>
<td></td>
<td>• MAC: MAC_ADDR updated — MAC addresses.</td>
</tr>
<tr>
<td></td>
<td>• Begin update ShortestPath rs with Wrap mode — Began updating the shortest path ringlet selection table in wrap mode.</td>
</tr>
<tr>
<td></td>
<td>• End update dynamic rs with Wrap mode — Ended updating the dynamic ringlet selection table in wrap mode.</td>
</tr>
<tr>
<td></td>
<td>• Begin update ShortestPath rs with Steer mode — Began updating the shortest path ringlet selection table in steer mode.</td>
</tr>
<tr>
<td></td>
<td>• End update dynamic rs with ShortestPath — Ended updating the dynamic ringlet selection table with ShortestPath.</td>
</tr>
<tr>
<td></td>
<td>• Update String RS start — Started updating the ringlet selection table. The String field can be default, shortest path, overall, VRRP, IPv6, VLANTunnel, or mac-learning.</td>
</tr>
<tr>
<td></td>
<td>• Update String RS finished — Finished updating the ringlet selection table. The String field can be default, shortest path, overall, VRRP, IPv6, VLANTunnel, or mac-learning.</td>
</tr>
<tr>
<td></td>
<td>• Update RS start — Began to update the ringlet selection table.</td>
</tr>
<tr>
<td></td>
<td>• Update RS finished — Finished updating the ringlet selection table.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RPR Event: On interface-type interface-number, String</th>
<th>The String field indicates the event that occurred:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• TP content change flag is set.</td>
</tr>
<tr>
<td></td>
<td>• Successed to Start protocol.</td>
</tr>
<tr>
<td></td>
<td>• Begin to delete Ipv6 Mac Address: MAC_ADDR.</td>
</tr>
<tr>
<td></td>
<td>• Begin to add Ipv6 Mac Address: MAC_ADDR.</td>
</tr>
<tr>
<td></td>
<td>• End to add Ipv6 Mac Address: MAC_ADDR.</td>
</tr>
</tbody>
</table>
### Field Description

- **RPR_Event:** On *interface-type* *interface-number*, at *String1* span, *String2*

  - *String1* field indicates the direction (east or west)
  - *String2* field indicates the event that occurred:
    - **Interface up**—An interface went up.
    - **Interface down**—An interface went down.
    - **Mistake cable is occurred**—Miscabling occurred.
    - **Protection status is changed**—Protection state changed.
    - **MATE status is changed**—The state of the MATE port changed.

### Examples

# Enable debugging for RPR ringlet selection event debugging. The output in this example was created when two stations form a closed RPR ring. When you remove the optical fiber to trigger protection on the station and update the ringlet selection table, output similar to the following example is generated:

```
<Sysname> debugging rpr event ringlet-selection
*Jun  5 11:24:51:218 2007 Sysname RPR/7/EVT_RS:Slot=3;
  RPR_Event: On RPR1, overall ringlet selection table was updated! Update overall RS start!
  // On interface RPR1, the overall ringlet selection table was updated. RPR began to update the overall ringlet selection table.
*Jun  5 11:24:51:889 2007 Sysname RPR/7/EVT_RS:Slot=3;
  RPR_Event: On RPR1, overall ringlet selection table was updated! Update overall RS finished!
  // On interface RPR1, the overall ringlet selection table was updated. RPR ended the updating of the overall ringlet selection table.
```

# Enable debugging for RPR ringlet selection event debugging on interface RPR1. The output in this example was created when two stations form a closed RPR ring. When you configure FS in the west direction of a station to change the topology and update the ringlet selection table, output similar to the following example is generated:

```
<Sysname> debugging rpr event ringlet-selection int rpr 1
*Jun  5 11:28:49:170 2007 Sysname RPR/7/EVT_RS:Slot=5;
  RPR_Event: On RPR1, dynamic ringlet selection table was updated! Update shortest path RS start!
  // On interface RPR1, the dynamic ringlet selection table was updated. RPR began the updating of the shortest path ringlet selection table.
*Jun  5 11:28:49:359 2007 Sysname RPR/7/EVT_RS:Slot=5;
  RPR_Event: On RPR1, dynamic ringlet selection table was updated! Update shortest path RS finished!
  // On interface RPR1, the dynamic ringlet selection table was updated. RPR finished the updating of the shortest path ringlet selection table.
```

# Enable RPR general event debugging. The output in this example was created when two stations form a closed RPR ring. When you shut down the west physical port, output similar to the following example is generated:

```
<Sysname> debugging rpr event general
*Jun  5 11:34:22:360 2007 Sysname RPR/7/EVT_SPAN:
  RPR_Event: On RPR1, at west span, Interface down.
  // On interface RPR1, the west physical port went down.
```
Enable RPR general event debugging. The output in this example was created when two stations form a closed RPR ring. When you remove the optical fiber connected to the west physical port, output similar to the following example is generated:

```
<Sysname> debugging rpr event general int rpr 1
*Jun 5 11:38:40:211 2007 Sysname RPR/7/EVT_SPAN:Slot=3;
  RPR_Event: On RPR1, at west span, Protection status is changed!
```

// On interface RPR1, the protection state on the west span changed.

designing rpr fsm

Use `designing rpr fsm` to enable RPR state machine debugging.

Use `undo designing rpr fsm` to disable RPR state machine debugging.

**Syntax**

```
designing rpr fsm [ interface interface-type interface-number ]
undo designing rpr fsm [ interface interface-type interface-number ]
```

**Default**

RPR state machine debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

`interface interface-type interface-number`: Specifies a port by its type and number.

**Usage guidelines**

Table 3 describes output fields and messages for the `designing rpr fsm` command.

**Table 177 Output from the designing rpr fsm command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPR_Fsm: On</td>
<td>The <code>String</code> field indicates the current protection event:</td>
</tr>
<tr>
<td>interface-type</td>
<td>• TP_RCVD—Received TP frames.</td>
</tr>
<tr>
<td>interface-number</td>
<td>• WTR_EXP—WTR timer expired.</td>
</tr>
<tr>
<td></td>
<td>• AUTO_IDLE—Link recovery event.</td>
</tr>
<tr>
<td></td>
<td>• AUTO_SD—Signal degraded.</td>
</tr>
<tr>
<td></td>
<td>• AUTO_SF—Signal failure.</td>
</tr>
<tr>
<td></td>
<td>• ADMIN_IDLE—Manual recovery.</td>
</tr>
<tr>
<td></td>
<td>• ADMIN_MS—Manual switch.</td>
</tr>
<tr>
<td></td>
<td>• ADMIN_FS—Forced switch.</td>
</tr>
</tbody>
</table>

312
### Field Description

The **String** field indicates the current state machine:
- **WTR state machine.**
- **receive TP frame state machine**—Receiving TP frames.
- **transmit TP frame state machine**—Transmitting TP frames.
- **topology DB update state machine** stage1—Stage 1 of topology database updating.
- **topology DB update state machine** stage2—Stage 2 of topology database updating.
- **protection state machine.**
- **topology validation state machine.**
- **TC frame processing state machine.**
- **ATD frame processing state machine**—ATD frame processing state machine.

### Examples

# Enable RRPP state machine debugging. The output in this example was created when two stations form a closed RPR ring. When you remove the optical fiber, output similar to the following example is generated:

```bash
<Sysname> debugging rpr fsm
*Jun  5 11:44:12:982 2007 Sysname RPR/7/FSM_PRTEVT:Slot=5;
RPR_Fsm: On RPR1, at west span, AUTO_SF protection switch is current.
```

// **On interface RPR1, signal failure occurred on the west span.**

```bash
*Jun  5 11:44:13:142 2007 Sysname RPR/7/FSM:Slot=5;
RPR_Fsm: On RPR1, at west span, protection state machine: in stage START
```

// **On interface RPR1, the protection state machine was in the START stage on the west span.**
# Enable RPR state machine debugging on interface RPR1. The output in this example was created when
two stations form a closed RPR ring. When you shut down the west physical port, output similar to the
following example is generated:

```plaintext
<Sysname> debugging rpr fsm interface rpr 1
*Jun 5 11:50:04:203 2007 Sysname RPR/7/FSM:Slot=5;
  RPR_Fsm: On RPR1, at west span, protection state machine: in stage POST

// On interface RPR1, the protection state machine was in the POST stage on the west span.
*Jun 5 11:50:04:363 2007 Sysname RPR/7/FSM:Slot=5;
  RPR_Fsm: On RPR1, at west span, protection state machine: in stage FINAL

// On interface RPR1, the protection state machine was in the FINAL stage on the west span.
```

debugging rpr packet

Use `debugging rpr packet` to enable debugging for RPR packets.

Use `undo debugging rpr packet` to disable debugging for RPR packets.

**Syntax**

```plaintext
debugging rpr packet [ [ atd | echo-request | echo-response | tc | tp ] [ receive | send ] ] [ tc | tp ] burst_send ] [ interface interface-type interface-number ] [ verbose ]
undo debugging rpr packet [ [ atd | echo-request | echo-response | tc | tp ] [ receive | send ] ] [ tc | tp ] burst_send ] [ interface interface-type interface-number ]
```

**Default**

The debugging for RPR packets is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- `atd`: Specifies debugging for ATD frames.
- `echo-request`: Specifies debugging for echo-request frames.
- `echo-response`: Specifies debugging for echo-response frames.
- `tc`: Specifies debugging for TC frames.
- `td`: Specifies debugging for TD frames.
- `receive`: Specifies debugging for received RPR packets.
- `send`: Specifies debugging for sent RPR packets.
- `burst_send`: Specifies debugging for burst_sent RPR packets.
- `verbose`: Displays the detailed information about RPR packets. If this keyword is not specified, the brief information about RPR packets is displayed.
- `interface interface-type interface-number`: Specifies a port by its type and number.
Usage guidelines

If no packet type is specified, debugging for all types of RPR packets is enabled.

If neither `receive` nor `send` is specified, debugging for both received and sent RPR packets is enabled.

**Table 4** describes output fields and messages for the `debugging rpr packet` command.

**Table 178 Output from the debugging rpr packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| RPR_Packet: On `interface-type interface-number`, at `String1` span, `String2` packet was `String3`. `String 4` | RPR packet debugging information:  
  • `String1`—Direction (east or west).  
  • `String2`—Packet type, which can be TP, TC, ATD, echo-response, or echo-request.  
  • `String3`—Received/sent packets. Received indicates the packets received, sent indicates the packets sent, and burst-sent indicates the packets burst-sent.  
  • `String4`—All contents of the packet. |

Examples

# Enable RPR packet debugging on the device and display brief RPR packet debugging information. The output in this example was created when two stations form a closed RPR ring.

```
<Sysname> debugging rpr packet
*Jun  5 12:43:02:365 2007 Sysname RPR/7/PKT_BRIEF_FUNC:Slot=5;
RPR_Packet: On RPR1, at east span, ATD packet was sent.
```

// On interface RPR1, the east span sent ATD frames.

# Enable RPR packet debugging on interface RPR1 and display detailed RPR packet debugging information. The output in this example was created when two stations form a closed RPR ring.

```
<Sysname> debugging rpr packet interface rpr 1 verbose
*Jun  5 12:44:08:365 2007 Sysname RPR/7/PKT_VERBOSE_FUNC:Slot=5;
RPR_Packet: On RPR1, at east span, ATD packet was sent.
ttl:255  ri:0  fe:0  ft:1  sc:3  we:0  parity:0
DA:ffff-ffff-SA:000f-e231-256d
ttlBase:255  ef:0  fi:0  ps:0  so:0  res:0
controlType:1  controlVersion:0
Ringlet0 weight: 0, ringlet1 weight: 0
Ringlet0 reserveband: 0, ringlet1 reserveband: 0
Station setting: multichoke-user 0;conversative 0;badfcs-user 0
Station name:
Manage address: 0.0.0.0
Ifindex: 63373312
Second mac1: 0000-0000-0000 Second mac2: 0000-0000-0000
```

// On interface RPR1, all contents of ATD frames sent to the east span were displayed.

# Enable debugging for RPR TP frames burst-sent on interface RPR1 and display detailed RPR packet debugging information. The output in this example was created when two stations form a closed RPR ring.

```
<Sysname> debugging rpr packet tp burst-send interface rpr 1 verbose
*Jun  5 13:45:21:56 2007 Sysname RPR/7/PKT_VERBOSE:Slot=5;
RPR_Packet: On RPR1, at west span, TP packet was burst-sent.
```
// On interface RPR1, all contents of TP frames burst-sent to the west span were displayed.

NOTE:
For TP frames and TC frames, the sending function is mutually exclusive with the burst-sending function.

debugging rpr timer

Use `debugging rpr timer` to enable RPR timer debugging.
Use `undo debugging rpr timer` to disable RPR timer debugging.

**Syntax**

```
debugging rpr timer [ interface interface-type interface-number ]
undo debugging rpr timer [ interface interface-type interface-number ]
```

**Default**

RPR timer debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

`interface interface-type interface-number`: Specifies a port by its type and number.

**Usage guidelines**

Table 5 describes output fields and messages for the `debugging rpr timer` command.
### Table 179 Output from the debugging rpr timer command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Timer event.</td>
</tr>
<tr>
<td></td>
<td>The String1 field indicates the timer name:</td>
</tr>
<tr>
<td></td>
<td>• TD-Fast—TD fast timer.</td>
</tr>
<tr>
<td></td>
<td>• TD-Slow—TD slow timer.</td>
</tr>
<tr>
<td></td>
<td>• TC-Fast—TC fast timer.</td>
</tr>
<tr>
<td></td>
<td>• TC-Slow—TC slow timer.</td>
</tr>
<tr>
<td></td>
<td>• ATD—ATD timer.</td>
</tr>
<tr>
<td></td>
<td>• holdoff—Holdoff timer.</td>
</tr>
<tr>
<td></td>
<td>• keepalive—Keepalive timer.</td>
</tr>
<tr>
<td></td>
<td>• stability—Stability timer.</td>
</tr>
<tr>
<td></td>
<td>• instability—Instability timer.</td>
</tr>
<tr>
<td></td>
<td>• Oam—OAM timer.</td>
</tr>
<tr>
<td></td>
<td>• IdleIsolateDetect—IdleIsolateDetect timer.</td>
</tr>
<tr>
<td></td>
<td>• ReportDefect—ReportDefect timer.</td>
</tr>
<tr>
<td></td>
<td>The String2 field indicates the timer action:</td>
</tr>
<tr>
<td></td>
<td>• starts.</td>
</tr>
<tr>
<td></td>
<td>• stops.</td>
</tr>
<tr>
<td></td>
<td>• expires.</td>
</tr>
</tbody>
</table>

#### Examples

# Enable RRPP timer debugging. The output in this example was created when two stations form a closed RPR ring. When you remove the main interface board to trigger protection on the station, output similar to the following example is generated:

```bash
< Sysname > debugging rpr timer
Jun 5 13:46:50:236 2007 Sysname RPR/7/TIMER:Slot=5;
RPR_Timer: On RPR1, instability timer starts.

// On interface RPR1, the instability timer started.
```

# Enable RRPP timer debugging on interface RPR1. The output in this example was created when two stations form a closed RPR ring. When you remove the optical fiber connected to the main interface to trigger protection on the station, output similar to the following example is generated:

```bash
< Sysname > debugging rpr timer interface rpr 1
Jun 5 13:46:50:536 2007 Sysname RPR/7/TIMER:Slot=5;
RPR_Timer: On RPR1, stability timer expires.

// On interface RPR1, the stability timer expired.
```
The output description tables in this document only contain fields and messages that require an explanation.

**debugging rrpp**

Use `debugging rrpp` to enable RRPP debugging.

Use `undo debugging rrpp` to disable RRPP debugging.

**Syntax**

```
debugging rrpp [ domain domain-id [ ring ring-id ] ] { all | error | event | fast-detect-fsm | fast-detect-packet | fsm | packet }
undo debugging rrpp [ domain domain-id [ ring ring-id ] ] { all | error | event | fast-detect-fsm | fast-detect-packet | fsm | packet }
```

**Default**

RRPP debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

`domain-id`: Specifies an RRPP domain by its ID, in the range of 1 to 8.

`ring-id`: Specifies an RRPP ring by its ID, in the range of 1 to 64.

`all`: Specifies all types of debugging for RRPP.

`error`: Specifies debugging for RRPP errors.

`event`: Specifies debugging for RRPP events.

`fast-detect-fsm`: Specifies debugging for the RRPP fast detection state machine.

`fast-detect-packet`: Specifies debugging for RRPP fast detection packets.

`fsm`: Specifies debugging for RRPP state machines.

`packets`: Specifies debugging for RPPP packets.

**Usage guidelines**

RRPP debugging can take effect on three levels: global, domain-level, and ring-level.

- If neither the `domain-id` argument nor the `ring-id` argument is specified, the operation takes effect globally. In this case, the `debugging rrpp` command enables debugging on all the rings and in all the domains.

- If only the `domain-id` argument is specified, the operation applies to all the rings in the domain.
- If both the `domain-id` argument and the `ring-id` argument are specified, the operation applies to the specified ring in the specified domain.

Table 1 describes output fields and messages for the `debugging rrpp error` command.

**Table 180 Output from the debugging rrpp error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Failed to String</strong></td>
<td>The memory operation failed. String field value:</td>
</tr>
</tbody>
</table>
|  | • `allocate memory`—Applying for dynamic memory failed.  
|  | • `copy MBUF to PDU`—Copying Ethernet network packets to the PDU failed. |
| **Domain domain-id Ring ring-id** | Errors occurred when a port sent/received packets. |
| **Port interface-type** | |
| **interface-number : String** | The string field describes the errors: |
|  | • `domain-id`—RRPP domain ID.  
|  | • `ring-id`—RRPP ring ID.  
|  | • `interface-type interface-number`—Interface type and interface number.  
|  | • The string field describes the errors:  
|  | o `Master node received Health packet from primary port`—The primary port of the master node received Health packets from the local node.  
|  | o `Send RRPP packet error`—Sending RRPP packets failed. |
| **Failed to write String queue** | RRPP failed to write a queue. String field value: |
|  | • `packet`—Writing a packet queue failed.  
|  | • `link status`—Writing a link status queue failed. |
| **Failed to write String event** | RRPP failed to write an event. String field value: |
|  | • `packet`—Writing a packet event failed.  
|  | • `link status`—Writing a link status event failed. |
A port received error packets. The reason is also given. The **string** field describes the errors:

- **illegal RRPP packet Length**—The RRPP packet length field in the RRPP packet received is invalid.
- **illegal RRPP packet version**—The RRPP version of the RRPP packet received is invalid.
- **illegal RRPP PDU Length**—The RRPP PDU length field in the RRPP packet received is invalid.
- **illegal domain ID**—The domain ID in the RRPP packet received is invalid.
- **inexistent domain**—The domain ID in the RRPP packet received was not the one configured on the local device.
- **no active domain**—The domain ID in the RRPP packet received was not activated, which means no ring was activated in the domain.
- **illegal level**—The level in the RRPP packet received is invalid.
- **illegal RRPP packet**—The type of the RRPP packet received is invalid.
- **packet receives from non-ctrlvlan**—The RRPP packet received was not from the control VLAN of the specified domain, which means the control VLAN did not match.
- **illegal ring ID**—The ring ID in the RRPP packet received is invalid.
- **hello time out of range**—The setting of the Hello timer in the RRPP packet received is out of range.
- **fail time out of range**—The setting of the Fail timer in the RRPP packet received is out of range.
- **value of fail-time must not be less than triple value of hello-time**—The fail timer value is less than three times the Hello timer value.
- **level mismatch**—The ring level in the RRPP packet received does not match the level of the ring in which the device resides.
- **a conflicting master node of current ring was detected**—Two master nodes existed on the ring (this message was output by the master node).

**Table 2** describes output fields and messages for the `debugging rppp event` command.

**Table 181 Output from the debugging rppp event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>domain-id</strong></td>
<td>domain-id indicates the RRPP domain ID.</td>
</tr>
<tr>
<td><strong>ring-id</strong></td>
<td>ring-id indicates the RRPP ring ID.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td><strong>is activated</strong>—The ring is activated.</td>
</tr>
<tr>
<td></td>
<td><strong>is inactivated</strong>—The ring is deactivated.</td>
</tr>
<tr>
<td></td>
<td><strong>turns to fault for link down</strong>—The ring fails because the link is down.</td>
</tr>
<tr>
<td></td>
<td><strong>turns to fault for Link-Down packet</strong>—The ring fails because a link-down</td>
</tr>
<tr>
<td></td>
<td>packet is received.</td>
</tr>
<tr>
<td></td>
<td><strong>turns to fault for fail-timer timeout</strong>—The ring fails because the master</td>
</tr>
<tr>
<td></td>
<td>node receives no health packet of its own when the Fail timer expires.</td>
</tr>
<tr>
<td></td>
<td><strong>recovered for health packet</strong>—The ring recovers because the master node</td>
</tr>
<tr>
<td></td>
<td>receives health packets of its own.</td>
</tr>
</tbody>
</table>

320
Table 3 describes output fields and messages for the `debugging rppp fast-detect-fsm` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>State machine information of a specific RRPP ring in a specific RRPP domain.</td>
</tr>
<tr>
<td>domain-id</td>
<td>RRPP domain ID.</td>
</tr>
<tr>
<td>ring-id</td>
<td>RRPP ring ID.</td>
</tr>
<tr>
<td>String field value:</td>
<td></td>
</tr>
<tr>
<td>RX FSM</td>
<td>Receive state machine.</td>
</tr>
<tr>
<td>TX FSM</td>
<td>Transmit state machine.</td>
</tr>
<tr>
<td>RXTX FSM</td>
<td>Receive and transmit state machines.</td>
</tr>
<tr>
<td>Previous state is State</td>
<td>Previous state of the state machine: Idle, Active, Complete, or Failed.</td>
</tr>
<tr>
<td>Current state is State</td>
<td>Current state of the state machine: Idle, Active, Complete, or Failed.</td>
</tr>
<tr>
<td>Transition event is String</td>
<td>The String field indicates the state transition conditions:</td>
</tr>
<tr>
<td></td>
<td>• Receiving Fast-Detect packet from the primary port — The primary port received fast-detection packets.</td>
</tr>
<tr>
<td></td>
<td>• Receiving Fast-Detect packet from the secondary port — The secondary port received fast-detection packets.</td>
</tr>
<tr>
<td></td>
<td>• Fast-FailTimer-Expired — The Fast-Fail Timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Fail-Timer-Expired — The Fail timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Detect-Enabled — Fast detection was enabled.</td>
</tr>
<tr>
<td></td>
<td>• Detect-Disabled — Fast detection was disabled.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the `debugging rppp fast-detect-packet` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Information about fast-detection packets:</td>
</tr>
<tr>
<td>domain-id</td>
<td>RRPP domain ID.</td>
</tr>
<tr>
<td>ring-id</td>
<td>RRPP ring ID.</td>
</tr>
<tr>
<td>String1</td>
<td>Rcvd for receiving packets and Send for sending packets.</td>
</tr>
<tr>
<td>String2</td>
<td>All contents of the packet in hexadecimal format.</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the `debugging rppp fsm` command.
Table 184 Output from the debugging rrpp fsm command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Domain domain-id ring-id String | Information about a specific state machine of a specific ring in a specific domain:  
  - **domain-id**—RRPP domain ID.  
  - **ring-id**—RRPP ring ID.  
  - The **String1** field represents the state machine type:  
    - **Master Node FSM**—Master node state machine.  
    - **Transit Node FSM**—Transit node state machine.  
    - **Edge Node FSM**—Edge node state machine.  
    - **Assistant-Edge Node FSM**—Assistant edge node state machine. |
| Previous state is State | Previous state of the state machine: **Complete**, **Failed**, **Init**, **Link-up**, **Link-Down**, **Preforwarding**, **Link-Up-Notify**, **Link-Down-Notify**, **Preforward-Notify**, or **Idle**. |
| Current state is State | Current state of the state machine: **Complete**, **Failed**, **Init**, **Link-up**, **Link-Down**, **Preforwarding**, **Link-Up-Notify**, **Link-Down-Notify**, **Preforward-Notify**, or **Idle**. |
| Transition event is String | A transition event occurred. The **String2** field gives the cause of the state transition:  
  - **Ring-Enabled**—The ring is enabled.  
  - **Ring-Disabled**—The ring is disabled.  
  - **the ports status under Init state**—The port is in the initial state.  
  - **Hello-Timer-Expired**—The Hello timer expires.  
  - **Fail-Timer-Expired**—The Fail timer expires.  
  - **Edgehello-Timer-Expired**—The Edgehello timer expires.  
  - **Edgefail-Timer-Expired**—The Edgefail timer expires.  
  - **Receiving own Health packet.**  
  - **Receiving Link-Down packet.**  
  - **Receiving Common-Flush-FDB packet.**  
  - **Receiving Complete-Flush-FDB packet.**  
  - **Receiving Sub-Ring-FDB packet.**  
  - **Receiving Edge-Hello packet.**  
  - **Receiving Major-Fault packet.**  
  - **own link down.**  
  - **own link restoring.** |

Table 6 describes output fields and messages for the **debugging rrpp packet** command.
Table 185 Output from the debugging rrpp packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port interface-type interface-number String1 packet of domain domain-id ringring-id.</td>
<td>(Length: ULONG, Type: String2) String3</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>Port interface-type interface-number String1 packet of domain domain-id ringring-id. (Length: ULONG, Type: String2) String3</td>
<td></td>
</tr>
</tbody>
</table>

Packet debugging information:
- **interface-type interface-number**—Interface type and interface number.
- **String1**—Packet direction. This field can be **Rcvd** (indicating received packets) or **Send** (indicating sent packets).
- **domain-id**—RRPP domain ID.
- **ring-id**—RRPP ring ID.
- **ULONG**—Packet length.
- **String2**—Packet type: **Health**, **Link-Down**, **Complete-Flush-FDB**, **Common-Flush-FDB**, **Edge-Hello**, or **Major-Fault**.
- **String3**—The packet content in hexadecimal format.

Examples

# Enable debugging for RRPP errors on one of the two master nodes. The output in this example was created when the following conditions exist:
- Configure two master nodes on an RRPP ring, and configure the other devices as the transit nodes.
- Enable RRPP on all the devices, and enable all the RRPP rings.

```
<Sysname> debugging rrpp error
*0.2484426 Sysname RRPP/8/RCVPKTERR:
Received packet on port Ethernet1/1 error ! Reason : a conflicting master node of current ring was detected .

*0.2485436 Sysname RRPP/8/RCVPKTERR:
Received packet on port Ethernet1/1 error ! Reason : a conflicting master node of current ring was detected .
```

// Two master nodes were detected on the ring.

# Enable debugging for RRPP events on Device B. The output in this example was created when Device A operates as a transit node on the primary ring, with the RRPP ring enabled and RRPP protocol disabled. When you configure Device B as the master node of the primary ring, with the RRPP ring and RRPP protocol enabled, output similar to the following example is generated:

```
<Sysname> debugging rrpp event
*0.6664449 Sysname RRPP/8/EVT:
Domain 1 ring 1 is activated .

*0.6667140 Sysname RRPP/8/EVT:
Domain 1 ring 1 turns to fault for fail-timer timeout .
```

// The ring failed because the master node received no health packet of its own before the Fail timer expired.

```
*0.8708437 Sysname RRPP/8/EVT:
Domain 1 ring 1 recovered for health packet .
```

// Ring 1 in domain 1 on Device B recovered after RRPP was enabled on Device A.

# Enable debugging for RRPP fast detection. The output in this example was created when transit nodes and the assistant-edge node are configured for the primary ring.

```
<Sysname> debugging rrpp fast-detect-fsm
```
Domain 1 ring 2 RX FSM. Previous state is Complete. Current state is Failed. Transition event is Fast-FailTimer-Expired.

// The receive state machine of ring 2 in RRPP domain 1 transited from Complete to Failed because the Fast-Fail timer expired.

# Enable debugging for fast-detection packets. The output in this example was created when the following conditions exist:

- A transit node is configured, which uses the default timer settings.
- Ethernet 1/1 is configured as the primary port.

<Sysname> debugging rrpp fast-detect-packet

Domain 1 ring 2 Send fast-detect packet.(Length: 98, count: 1)
00 0f e2 07 83 98 00 0f e2 03 fd 75 81 00 e9 58
7f fe 06 6f 79 f0 70 00 00 48 aa aa 03 00 e0 2b
00 bb 99 0b 00 40 01 0d 00 0b 00 02 00 00 00 00
00 01 77 77 00 01 00 03 00 01 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00

// Ring 2 in domain 11 sent fast-detection packets.

# Enable debugging for RRPP state machines on the device. The output in this example was created when a device is configured as the master node of the primary ring, with the following settings:

- The timers adopt the default settings.
- Ethernet 1/1 is configured as the primary port.
- Ethernet 1/2 is configured as the secondary port.

<Sysname> debugging rrpp fsm

Domain 1 ring 1 Master Node FSM. Previous state is Complete. Current state is Complete. Transition event is Hello-Timer-Expired.

// The information about the state machines of the master node of ring 1 in domain 1 was displayed. The previous state of the state machine was Complete, and the current state of the state machine was Complete. The state transited because the Hello timer expired.

*Sysname RRPP/7/RRPPFSM:

Domain 1 ring 1 Master Node FSM. Previous state is Complete. Current state is Complete. Transition event is Receiving own Health packet.

// The information about the state machines of the master node of ring 1 in domain 1 was displayed. The previous state of the state machine was Complete, and the current state of the state machine is Complete. The state transited because the master node received a health packet of its own.

# Enable debugging for RRPP packets on the device. The output in this example was created when a device is configured as the master node of the primary ring, with the following settings:

- The timers adopt the default settings.
- Ethernet 1/1 is configured as the primary port.
- Ethernet 1/2 is configured as the secondary port.

<Sysname> debugging rrpp packet

324
Port Ethernet1/1 Send packet of domain1 ring1.(Length: 64, Type: Health)
99 0b 00 40 01 05 00 01 00 01 00 00 00 00 00 00 00
01 11 00 01 00 03 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

// The primary port Ethernet 1/1 sent a Health packet.

*0.9497605 Sysname RRPP/8/RRPPKT:
Port Ethernet1/2 Rcvd packet of domain1 ring1.(Length: 64, Type: Health)
99 0b 00 40 01 05 00 01 00 01 00 00 00 00 00 00 00
01 11 00 01 00 03 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

// The secondary port Ethernet 1/2 received the Health packet the local node had sent.
RSH debugging commands

depending rsh

Use `debugging rsh` to enable RSH debugging.
Use `undo debugging rsh` to disable RSH debugging.

Syntax

`debugging rsh`
`undo debugging rsh`

Default

RSH debugging is disabled.

Views

User view

Default command level

1: Monitor level

Examples

# Enable RSH debugging. When RSH is used to execute an OS command of a remote host, output
similar to the following example is generated:

```
<Sysname> debugging rsh
<Sysname> rsh 1.1.1.1 command cd c:
Trying 1.1.1.1 ...
Press CTRL+K to abort
*0.5850881 Sysname RSH/8/REQUEST:User Sysname start connection to Server (1.1.1.1)
*0.5850990 Sysname RSH/8/SEND:User Sysname sent 23 byte(s) to server (1.1.1.1)
*0.5851090 Sysname RSH/8/RECV:User Sysname received 1 byte(s) from server (1.1.1.1)
*0.5851200 Sysname RSH/8/CLOSECONN:Server (1.1.1.1) closed connection
```

// The device initiated an RSH connection to remote host 1.1.1.1 (the RSH server), sent 23 bytes, and
received 1 byte. The remote host closed the RSH connection.
Service loopback group debugging commands

The service loopback group module name is identified as “SLBG” in debugging messages.

**debugging service-loopback**

Use **debugging service-loopback** to enable service loopback group debugging. Use **undo debugging service-loopback** to disable service loopback group debugging.

**Syntax**

```
debugging service-loopback { all | error | event }
undo debugging service-loopback { all | error | event }
```

**Default**

Service loopback group debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor Level

**Parameters**

- **all**: Specifies all types of service loopback group debugging.
- **error**: Specifies service loopback group error debugging.
- **event**: Specifies service loopback group event debugging.

**Examples**

# Enable service loopback group error debugging. When SLBG is modifying the service type on an interface card, output similar to the following example is generated:
```
<Sysname> debugging service-loopback error
*Nov  3 19:29:12:860 2007 Sysname SLBG/7/SLBG_DEBUG:
  EVENT.0-16338762:
  The service loopback group did not exist when modifying service type on IO board.
```

// The service loopback group did not exist.

# Enable service loopback group event debugging. When a service loopback group is removed, output similar to the following example is generated:
```
<Sysname> debugging service-loopback event
<Sysname> system-view
<Sysname> undo service-loopback group 1
*Nov  3 19:29:12:860 2007 Sysname SLBG/7/SLBG_DEBUG:
  EVENT.0-16338762:
  Service loopback group 1 is reporting message to another module.
```

// When service loopback group 1 was removed, the group reported the event to another module.
The output description tables in this document only contain fields and messages that require an explanation.

**debugging session aging-process**

Use `debugging session aging-process` to enable debugging for session aging queue processing of session management.

Use `undo debugging session aging-process` to disable debugging for session aging queue processing of session management.

**Syntax**

```
debugging session aging-process [ acl acl-number ]
undo debugging session aging-process
```

**Default**

Debugging for session aging queue processing of session management is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- `acl acl-number`: Uses an ACL to match sessions for logging. The `acl-number` argument specifies an ACL number in the range of 2000 to 3999. If you execute the command multiple times with different ACLs, the most recent command takes effect.

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging session aging-process` command.
Table 186 Output from the debugging session aging-process command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| State: PRO_STATE | Protocol states:  
|                | • TCP_SYN_OPEN.  
|                | • TCP_ESTABLISHED.  
|                | • TCP_FIN_CLOSE.  
|                | • TCP_CLOSED.  
|                | • UDP_OPEN.  
|                | • UDP_READY.  
|                | • ICMP_REQUEST.  
|                | • ICMP_REPLY.  
|                | • RAWIP_OPEN.  
|                | • RAWIP_READY.  
|                | • ICMP_ERROR_CTRL.  
|                | • PRO_FTP.  
|                | • PRO_DNS.  
|                | • PRO_MSN.  
|                | • PRO_QQ.  
| Session extension state: |  
| Extent-State | Session extension state:  
|              | • WAITACK0.  
|              | • WAITACK1.  
|              | • WAITACK2.  
|              | • INACTIVE.  
|              | • ACCELERATE.  
|              | • HALF_SESSION.  
|              | • KEEP_ALIVE.  
|              | • PERSIST.  
|              | • DYNAMIC.  

Examples

# Enable debugging for session aging queue processing on the device enabled with the security module. When the device pings 192.168.1.58, output similar to the following example is generated:

<Sysname> debugging session aging-process
<Sysname> ping 192.168.1.58

*May 27 10:30:28:846 2009 Sysname SESSION/7/AGINGPROC:
Tuple5: 3.3.3.2/2048-->3.3.3.1/3(ICMP)
State: ICMP_REQUEST Extent-State: NULL

// Session management detected that the protocol state of the ICMP session is ICMP_REQUEST, with no extension state.

debugging session engine

Use **debugging session engine** to enable debugging for session engine.

Use **undo debugging session engine** to disable debugging for session engine.
Syntax

debugging session engine { all | event | error }
undo debugging session engine { all | event | error }

Default

Debugging for session engine is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all: Specifies all types of debugging for session engine.

event: Specifies debugging for session engine events.

error: Specifies debugging for session engine errors.

Usage guidelines

Table 2 describes output fields and messages for the debugging session engine event command.

Table 187 Output from the debugging session engine event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Received msg message from SESSION</td>
<td>msg: Message name.</td>
</tr>
<tr>
<td></td>
<td>• SESSION_END_NOTIFY.</td>
</tr>
<tr>
<td></td>
<td>• SESSION_DELETE_NOTIFY.</td>
</tr>
<tr>
<td></td>
<td>• SESSION_ADD_RELATION_NOTIFY.</td>
</tr>
<tr>
<td></td>
<td>• SESSION_DEL_RELATION_NOTIFY.</td>
</tr>
<tr>
<td>Succeeded in notifying DRIVER to process the event: event</td>
<td>The session engine received a message from the session management module, and notified the driver to perform the processing action specified by event:</td>
</tr>
<tr>
<td></td>
<td>• A session was created.</td>
</tr>
<tr>
<td></td>
<td>• A session was deleted.</td>
</tr>
<tr>
<td></td>
<td>• A session was updated.</td>
</tr>
<tr>
<td></td>
<td>• A relation-table was created.</td>
</tr>
<tr>
<td></td>
<td>• A relation-table was deleted.</td>
</tr>
<tr>
<td>DRIVER refused to process the event from SESSION: event</td>
<td>The session engine received a message about the refuse event from the driver.</td>
</tr>
<tr>
<td></td>
<td>event: Event that the driver refused to process.</td>
</tr>
<tr>
<td>msg message not sent to DRIVER</td>
<td>The session engine did not send the message to the driver.</td>
</tr>
<tr>
<td></td>
<td>msg: Message name.</td>
</tr>
<tr>
<td></td>
<td>• SESSION_END_NOTIFY.</td>
</tr>
<tr>
<td></td>
<td>• SESSION_DELETE_NOTIFY.</td>
</tr>
<tr>
<td></td>
<td>• SESSION_ADD_RELATION_NOTIFY.</td>
</tr>
</tbody>
</table>
### Field Description

**Received message from DRIVER:** `msg`

- `Msg: Message name.`
  - `CREAT_SESSION.`
  - `UPDATE_SESSION.`
  - `DELETE_SESSION.`

**Succeeded in process according to DRIVER message**

- `process: Processing action.`
  - creating a session.
  - updating a session.
  - deleting a session.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pro</td>
<td>Session protocol.</td>
</tr>
<tr>
<td>ID</td>
<td>Session ID.</td>
</tr>
<tr>
<td>VPN Index</td>
<td>VPN index in the HASH entry.</td>
</tr>
<tr>
<td><code>Init IP/Port -&gt; resp IP/Port VpnIndex</code></td>
<td>HASH entry: Initiator IP/port -&gt; Responder IP/port VPN index.</td>
</tr>
<tr>
<td><code>Init IP/Port &lt;- resp IP/Port VpnIndex</code></td>
<td>Inverse HASH entry. Initiator IP/port &lt;- Responder IP/port VPN index.</td>
</tr>
</tbody>
</table>

**Table 3** describes output fields and messages for the **debugging session engine error** command.

**Table 188 Output from the debugging session engine error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVER failed to process the event: <code>event</code></td>
<td>The driver failed to process the message sent by the session engine, and the driver notified the session engine of the event. <code>event: Message name.</code></td>
</tr>
</tbody>
</table>
  - A session was created.
  - A session was deleted.
  - A session was updated.
  - A relation-table was created.
  - A relation-table was deleted.

<table>
<thead>
<tr>
<th>Failed to process according to DRIVER message</th>
<th>process: Processing action.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>creating a session.</td>
</tr>
<tr>
<td></td>
<td>updating a session.</td>
</tr>
<tr>
<td></td>
<td>deleting a session.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable debugging for session engine on a firewall device. When the device sends out a UDP packet, output similar to the following example is generated:

```
<Sysname> debugging session engine event
<Sysname> terminal debugging
*Apr 26 14:41:36:516 2007 Sysname SESS_DP/7/ENGINE: 
EVENT:Received SESSION_END_NOTIFY message from SESSION.
Pro:UDP  ID:57
Initiator: IP/ Port Responder: IP/ Port VPN Index
1.1.1.2/ 100 --> 2.2.2.2/ 100 0
```

331
The session engine received a SESSION_END_NOTIFY message sent by the session management module.

**Apr 26 14:41:36:517 2007 Sysname SESS_DP7/ENGINE:

EVENT: Succeeded in notifying DRIVER to process the event: A session was created.
Pro:UDP  ID:57
Initiator: IP/ Port Responder: IP/ Port VPN Index
1.1.1.2/ 100 --> 2.2.2.2/ 100 0
1.1.1.2/ 100 <-- 2.2.2.2/ 100 0

The session engine received a message from the driver. The driver created a session successfully.

Enable debugging for session engine on a firewall device. When the device sends out a TCP packet, output similar to the following example is generated:

<Sysname> debugging session engine error
<Sysname> terminal debugging

**Apr 26 14:41:36:516 2007 Sysname SESS_DP7/ENGINE:

EVENT: Received SESSION_END_NOTIFY message from SESSION.
Pro:TCP  ID:60
Initiator: IP/ Port Responder: IP/ Port VPN Index
1.1.1.2/ 100 --> 2.2.2.2/ 100 0
1.1.1.2/ 100 <-- 2.2.2.2/ 100 0

The session engine received a SESSION_END_NOTIFY message sent by the session management module.

**Apr 26 14:41:36:517 2007 Sysname SESS_DP7/ENGINE:

EVENT: DRIVER failed to process the event: A session was created.
Pro:TCP  ID:60
Initiator: IP/ Port Responder: IP/ Port VPN Index
1.1.1.2/ 100 --> 2.2.2.2/ 100 0
1.1.1.2/ 100 <-- 2.2.2.2/ 100 0

The session engine received a message from the driver. The driver failed to create a session.

debugging session ext-info

Use **debugging session ext-info** to enable debugging for extended information of session management.

Use **undo debugging session ext-info** to disable debugging for extended information of session management.

**Syntax**

debugging session ext-info { all | event | error } [ acl acl-number ]
undo debugging session ext-info { all | event | error }

**Default**

Debugging for extended information of session management is disabled.

**Views**

User view
Default command level

1: Monitor level

Parameters

**all**: Specifies all types of debugging for extended information.

**event**: Specifies debugging for events of extended information.

**error**: Specifies debugging for errors of extended information.

**acl acl-number**: Uses an ACL to match sessions for logging. The *acl-number* argument specifies an ACL number in the range of 2000 to 3999. If you execute the command multiple times with different ACLs, the most recent command takes effect.

Usage guidelines

Table 4 describes output fields and messages for the `debugging session ext-info event` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach, Detach, GetAttach</td>
<td>Extended information operation type: Attach, Detach, or GetAttach.</td>
</tr>
<tr>
<td>succeed</td>
<td>Extension information operation is done successfully.</td>
</tr>
</tbody>
</table>

Service modules:
- NAT.
- ASPF.
- ALG.
- STAT (attack prevention)
- TCPPROXY.
- ENGINE (session engine)
- P2P.
- LB.
- FLOW_REDIRECT.
- FLT6.
- NATPT.
- CONNLMT.
- PBR (policy routing)
- DDOS.
- SRVASST (Server Assistant)
- SESSIONLOG.

Examples

# Enable debugging for extended information on the device with the security module enabled. The output in this example was created when the following conditions exist:

- The NAT server function is configured.
- An ICMP packet that can be translated by the NAT server is sent.

```
<Sysname> debugging session ext-info all
*Mar 24 18:15:47:164 2007 Sysname DPSESSIO/7/EXTINFO:
   Attach succeed
Module NAT
```
debugging session packet-process

Use `debugging session packet-process` to enable debugging for packet processing of session management.

Use `undo debugging session packet-process` to disable debugging for packet processing of session management.

Syntax

```
debbuging session packet-process
undo debugging session packet-process
```

Default

Debugging for packet processing of session management is disabled.

Views

User view

Default command level

1: Monitor level

Parameters


Examples

```
# Enable debugging for packet processing on the device with the security module enabled. Output similar to the following example is generated when a checksum error packet is sent to the device under the condition that the device is enabled with checksum verification.
<Sysname> debugging session packet-process
<Sysname> system-view
[Sysname] session checksum all
*Mar 26 08:50:24:568 2007 Sysname DPSESSIO/7/PKTPROC:
  Tuple3: 192.168.1.58-->192.168.1.11(TCP)
  Received: checksum error packet
  // The packet process received a checksum error TCP packet.
*Mar 26 08:57:24:896 2007 Sysname DPSESSIO/7/PKTPROC:
  Tuple3: 192.168.0.21-->192.168.1.89(TCP)
  ERROR: unknown ICMP error control packet
  // The packet process received a TCP ICMP error control packet.
```

debbuging session relation

Use `debugging session relation` to enable debugging for session relation table of session management.

Use `undo debugging session relation` to disable debugging for session relation table of session management.
Syntax

debugging session relation { all | event | error }
undo debugging session relation { all | event | error }

Default

Debugging for session relation table of session management is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all: Specifies all types of debugging for session relation table.

event: Specifies debugging for session relation table events.

error: Specifies debugging for session relation table errors.

Usage guidelines

Table 5 describes output fields and messages for the debugging session relation event command.

Table 190 Output from the debugging session relation event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocalTuple3: localIP / localPort</td>
<td>Internal network 3-tuple of relation table.</td>
</tr>
<tr>
<td>GlobalTuple3: globalIP/globalPort(ProtoType)</td>
<td>External network 3-tuple of relation table.</td>
</tr>
<tr>
<td>Create, Delete, Update</td>
<td>Relation table events.</td>
</tr>
<tr>
<td>module call, time out, child full, module call / child full</td>
<td>Event causes.</td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the debugging session relation error command.

Table 191 Output from the debugging session relation error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error:</td>
<td>Relation table error.</td>
</tr>
<tr>
<td>Memory not enough for relation</td>
<td>Insufficient memory for creating a relation entry.</td>
</tr>
<tr>
<td>Relation number exceed max</td>
<td>The number of relation table entries exceeds the upper limit.</td>
</tr>
</tbody>
</table>

Examples

# Enable debugging for session relation table on the device with the security module enabled. When the ALG module added a relation table entry, output similar to the following example is generated:

```bash
<Sysname> debugging session relation all
*Mar 26 09:12:33:800 2007 Sysname DPSESSION/7/RELATION:
Create module call
// Session management created a relation entry after receiving a creation command from an external module.
```
debugging session session-table

Use `debugging session session-table` to enable debugging for session table entries of session management.

Use `undo debugging session session-table` to disable debugging for session table entries of session management.

Syntax

```
debugging session session-table { all | event | error | fsm } { acl acl-number }
undo debugging session session-table { all | event | error | fsm }
```

Default

Debugging for session table entries is disabled.

Views

User view

Default command level

2: System level

Parameters

- `all`: Specifies all types of debugging for session table.
- `event`: Specifies debugging for session table events.
- `error`: Specifies debugging for session table errors.
- `fsm`: Specifies debugging for session table state machine.
- `acl acl-number`: Uses an ACL to match sessions for logging. The `acl-number` argument specifies an ACL number in the range of 2000 to 3999. If you execute the command multiple times with different ACLs, the most recent command takes effect.

Usage guidelines

Table 7 describes output fields and messages for the `debugging session session-table fsm` command.

**Table 192 Output from the debugging session session-table fsm command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuple5(FSM):</td>
<td>Session 5-tuple.</td>
</tr>
<tr>
<td>FSM: preState--&gt;nextState,</td>
<td>State machine changes from one state to another state.</td>
</tr>
<tr>
<td>received: ProtoType</td>
<td>Transport layer protocol type of the received packet.</td>
</tr>
</tbody>
</table>
Examples

# Enable debugging for session table entries on the device with security module enabled. When a ping packet passes through the device, output similar to the following example is generated:

<Sysname> debugging session session-table all

*Mar 24 18:15:47:164 2007 Sysname DPSESSIO/7/TABLE:
   Tuple5(EVENT): 192.168.0.2/8-->192.168.1.58/3840(ICMP)
   Operation:Create

// Session management created an ICMP session entry.

*Mar 24 18:15:47:174 2007 Sysname DPSESSIO/7/TABLE:
   Tuple5(FSM): 192.168.0.2/8-->192.168.1.58/3840(ICMP)
   FSM:NONE --> READY ,received:ICMP

// Session management changed the session state from NONE to READY because of the receipt of ICMP packet.

   Tuple5(EVENT): 192.168.0.2/8-->192.168.1.58/4096(ICMP)
   Operation:Delete

// Session management deleted the session entry with source IP address 192.168.0.2 and destination IP address 192.168.1.58.

# Enable debugging for session table entries on the device with security module enabled. When the resources for session table are insufficient, output similar to the following example is generated:

   Error: Memory not enough for session

// Session management failed to apply for session table resources due to insufficient memory.

debugging session log proc

Use `debugging session log proc` to enable debugging for session log processing.

Use `undo debugging session log proc` to disable debugging for session log processing.

Syntax

```
debugging session log proc { all | error | event }
undo debugging session log proc { all | error | event }
```

Default

Debugging for session log processing is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- **all**: Specifies all types of debugging for session log processing.
- **error**: Specifies for session log processing errors.
- **event**: Specifies for session log processing events.
Examples

# Enable debugging for session log processing events on the device that supports session management.
<Sysname> debugging session log proc event

# Configure session logging output thresholds.
<Sysname> system-view
[Sysname] session log time-active 10
[Sysname] session log packets-active 1

# Enable the session logging function.
[Sysname] interface gigabitethernet 1/1
[Sysname-GigabitEthernet1/1] session log enable inbound

# Ping the device from 192.168.0.2. If the ping operation succeeds, output similar to the following messages is generated:
*Mar 26 08:50:24:568 2007 Sysname DPSESS/7/logprocess:
  Tuple5: 192.168.0.2/8-->192.168.1.58/3840(ICMP)
  Event: Logged a session creation event.

  // Session management created an ICMP session. The system logged the session creation event.

# Remove the ICMP session. When the ICMP session is removed successfully, output similar to the following messages is generated:
<Sysname> reset session source-ip 192.168.0.2
*Mar 26 08:50:24:568 2007 Sysname DPSESS/7/logprocess:
  Tuple5: 192.168.0.2/8-->192.168.1.58/3840(ICMP)
  Event: Logged a session deleted event, type is AGING.

  // Session management deleted the target ICMP session entry. The system logged the session deletion event, and the log type is AGING.

Output similar to the following messages is generated when the ICMP session lasts 10 minutes under the condition that the device is pinged continuously:
*Mar 26 08:50:24:568 2007 Sysname DPSESS/7/logprocess:
  Tuple5: 192.168.0.2/8-->192.168.1.58/3840(ICMP)
  Event: Logged a session hold time threshold reached event.

  // The system output a log about the event that the session reached the session hold time threshold.

Output similar to the following messages is generated when the number of ICMP session packets reaches 1M under the condition that the device is pinged continuously:
*Mar 26 08:50:24:568 2007 Sysname DPSESS/7/logprocess:
  Tuple5: 192.168.0.2/8-->192.168.1.58/3840(ICMP)
  Event: Logged a session traffic threshold reached event.

  // The system output a log about the event that the session reached the traffic threshold.

# Enable debugging for session log processing errors. The output in this example was created when the following conditions exist:
- The session logging function is enabled.
- Thresholds for outputting session logs are configured.
<Sysname> debugging session log proc error
*Dec 16 15:52:55:677 2007 Sysname Sysname/7/LOG_PROCESS:Slot=1;
Tuple5(EVENT): 192.168.119.119/8-->2.2.2.2/1280(ICMP)
  Error: Failed to log a session created event.
debugging session log active-flow

Use `debugging session log active-flow` to enable debugging for active flow.
Use `undo debugging session log active-flow` to disable debugging for active flow.

**Syntax**

```
debugging session log active-flow { all | event | error }
undo debugging session log active-flow { all | event | error }
```

**Default**

Debugging for active flow is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- `all`: Specifies all types of debugging for active flow.
- `event`: Specifies debugging for active flow operation events.
- `error`: Specifies debugging for active flow operation errors.

**Examples**

# Enable debugging for active flow logging events on the device that supports session management.
```bash
<Sysname> debugging session log active-flow event
```

# Configure session logging output thresholds.
```bash
<Sysname> system-view
[Sysname] session log time-active 10
```

# Enable the session logging function.
```bash
[Sysname] interface gigabitethernet 1/1
[Sysname-GigabitEthernet1/1] session log enable inbound
```

When a data flow passes through the device, output similar to the following example is generated:
```
*Mar 26 08:50:24:568 2007 Sysname DPSESS/7/active-flow:
  Event: Active-flow queue created.
```

// **Session management created an active-flow queue.**

# Disable the session logging function.
```bash
[Sysname-GigabitEthernet0/1] undo session log enable
```

```
*Mar 26 08:50:24:568 2007 Sysname DPSESS/7/active-flow:
  Event: Active-flow queue deleted.
```

// **Session management deleted the active-flow queue.**

# Ping the device from the host at 192.168.0.2. If the ping operation succeeds, output similar to the following messages is generated:
```
*Mar 26 08:50:24:568 2007 Sysname DPSESS/7/active-flow:
  Tuple5: 192.168.0.2/8--->192.168.1.58/3840(ICMP)
  Event: Session appended to the active-flow queue.
```
// Session management added the created ICMP session to the active-flow queue.

# Delete the created ICMP session. When the created ICMP session is deleted successfully, output similar to the following messages is generated:

<pre>
<Sysname> reset session source-ip 192.168.0.2
*Mar 26 08:50:24:568 2007 Sysname DPSESS/7/active-flow:
  TupleS: 192.168.0.2/8 --> 192.168.1.58/3840 (ICMP)
  Event: Session removed from the active-flow queue.

// Session management deleted the ICMP session from the active-flow queue.

# Enable debugging for active flow logging errors on a device that supports session management. The output in this example was created when the following conditions exist:

- The session logging function is enabled.
- Thresholds for outputting session logs are configured.

<pre>
<Sysname> debugging session log active-flow error
*Dec 16 15:55:35:377 2007 Sysname SESSION/7/LOG.ACTFLOW:Slot=2;
  Error: Failed to create the active-flow queue.
SIP debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

**debugging voice sip**

Use `debugging voice sip` to enable SIP debugging.

Use `undo debugging voice sip` to disable SIP debugging.

**Syntax**

```
debugging voice sip { all | error | event | message | stack | timer }
undo debugging voice sip { all | error | event | message | stack | timer }
```

**Default**

SIP debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **all**: Specifies all types of SIP debugging.
- **error**: Specifies error debugging.
- **event**: Specifies event debugging.
- **message**: Specifies message debugging.
- **stack**: Specifies protocol stack debugging.
- **timer**: Specifies timer debugging.

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging voice sip error` command.

**Table 193 Output from the debugging voice sip error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filling ACCP head failed</td>
<td>SIP failed to fill ACCP header.</td>
</tr>
<tr>
<td>Failed to Malloc Memory for XXX</td>
<td>SIP failed to allocate memory to XXX.</td>
</tr>
<tr>
<td>Get Local Address From Uri Failed</td>
<td>SIP failed to obtain the local address from URI.</td>
</tr>
<tr>
<td>The UserInfo Number is not exist</td>
<td>The UserInfo Number did not exist.</td>
</tr>
<tr>
<td>Send accp service ack msg failed</td>
<td>SIP failed to send an ACCP service Ack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Refer Response</td>
<td>Adapter module received a REFER response from the protocol stack.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Send Prack Request Failed</td>
<td>SIP failed to send a Prack request.</td>
</tr>
<tr>
<td>Send AccpAlerting message Failed</td>
<td>SIP failed to send an AccpAlerting message.</td>
</tr>
<tr>
<td>Send AccpConnect Failed</td>
<td>SIP failed to send an AccpConnect message.</td>
</tr>
<tr>
<td>Send SipConnectAck Failed</td>
<td>SIP failed to send a SipConnectAck message.</td>
</tr>
<tr>
<td>Find XXX error</td>
<td>SIP failed to get the XXX field.</td>
</tr>
<tr>
<td>Create AppRspMsg error</td>
<td>SIP failed to create an AppRspMsg message.</td>
</tr>
<tr>
<td>Cannot find the exist SIP CCB</td>
<td>The existing SIP CCB was not found.</td>
</tr>
<tr>
<td>SIP receive setup callinfotab error</td>
<td>SIP received a Setup message with an invalid call info table.</td>
</tr>
<tr>
<td>The CMC state of ccb is error</td>
<td>CMC state error in SIP CCB.</td>
</tr>
<tr>
<td>SIP Find codec failed</td>
<td>SIP failed to find the codec type.</td>
</tr>
<tr>
<td>Create MediaChannel Failed</td>
<td>SIP failed to create a media channel.</td>
</tr>
<tr>
<td>VIM Attach IppId(= %d) to Line(= %d) Failed</td>
<td>SIP failed to attach CCB of the IPP module to VIM.</td>
</tr>
<tr>
<td>VIM Detach IppId from Line(= %d) Failed</td>
<td>SIP failed to detach CCB of the IPP module from VIM.</td>
</tr>
<tr>
<td>Transfer User Number is empty</td>
<td>The transfer destination number was empty.</td>
</tr>
<tr>
<td>SIP TPTD module register high call back function failed</td>
<td>TPTD module failed to register a callback function with the higher layer.</td>
</tr>
<tr>
<td>Receive Network error packet</td>
<td>SIP received an error packet from the network side.</td>
</tr>
<tr>
<td>Receive Network too long packet</td>
<td>SIP received too long a packet from the network side.</td>
</tr>
<tr>
<td>Send udp Msg failed</td>
<td>SIP failed to send a UDP message.</td>
</tr>
<tr>
<td>Fail to create signal transport</td>
<td>SIP failed to create a signaling channel.</td>
</tr>
<tr>
<td>The contact header isn’t exist</td>
<td>The Contact header didn’t exist.</td>
</tr>
<tr>
<td>The number of contact item is invalid</td>
<td>The number of Contact headers was invalid.</td>
</tr>
<tr>
<td>Malloc for Require Failed</td>
<td>SIP failed to allocate memory to Require.</td>
</tr>
<tr>
<td>Encode SipFrag Failed</td>
<td>SIP failed to encode SipFrag.</td>
</tr>
<tr>
<td>Change Media Ip Addr Failed</td>
<td>SIP failed to change the IP address of the media channel.</td>
</tr>
<tr>
<td>Replaces Parameter is not exist in SIP CCB</td>
<td>The Replaces parameter didn’t exist in SIP CCB.</td>
</tr>
<tr>
<td>Analyze the destination of INVITE request: The SIP CCB has been existed.</td>
<td>Analyzing the destination of INVITE request: The SIP CCB existed.</td>
</tr>
<tr>
<td>Analyze the destination of INVITE request: SIP receive setup callinfotab error.</td>
<td>Analyzing the destination of INVITE request: SIP received a wrong callinfotab.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Failed to get authentication because failed to calculation response code.</td>
<td>SIP failed to obtain authentication information because it failed to calculate response code.</td>
</tr>
<tr>
<td>Failed to calculate authentication because the number of authentication header is negative.</td>
<td>SIP failed to calculate authentication information because the number of authentication headers was negative.</td>
</tr>
<tr>
<td>Build Options: Failed add To IE to Msg.</td>
<td>Build Options: SIP failed to add To IE to Msg.</td>
</tr>
<tr>
<td>Build Options: Failed add contact IE to msg.</td>
<td>Build Options: SIP failed to add contact IE to msg.</td>
</tr>
<tr>
<td>Delete server group: Failed to delete server %u, for it’s referenced.</td>
<td>Delete server group: SIP failed to delete server %u because it was referenced.</td>
</tr>
<tr>
<td>Failed to create the send options timer.</td>
<td>SIP failed to create the timer for sending OPTIONS messages.</td>
</tr>
<tr>
<td>Failed to create the highest priority server’s auxiliary timer.</td>
<td>SIP failed to create the auxiliary timer for the highest priority server.</td>
</tr>
<tr>
<td>OPTIONS Message Response: Message response is not with the right sequence.</td>
<td>OPTIONS Message Response: Message response had a wrong sequence number.</td>
</tr>
<tr>
<td>OPTIONS Message Response: Message response is not corresponding to the server using.</td>
<td>OPTIONS Message Response: Message response did not match the server in use.</td>
</tr>
<tr>
<td>Options DNS proc: DNS query failed.</td>
<td>Options DNS processing: DNS query failed.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the **debugging voice sip event** command.

### Table 194 Output from the debugging voice sip event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP --&gt; CMC : Accp Setup</td>
<td>SIP sent an ACCP Setup message to CMC.</td>
</tr>
<tr>
<td>SIP --&gt; CMC : XXX</td>
<td>SIP sent a XXX message to CMC.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : DIM Response</td>
<td>Adapter layer received a response from the DIM module of the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : DIM Request</td>
<td>Adapter layer received a request from the DIM module of the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Ssn Response</td>
<td>Adapter layer received a response from the SSN module of the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Ssn Request</td>
<td>Adapter layer received a request from the SSN module of the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Subscription Response</td>
<td>Adapter layer received a response from the Subscription module of the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Invite Request</td>
<td>Adapter layer received an Invite request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Refer Request</td>
<td>Adapter layer received a Refer request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Alerting Indication</td>
<td>Adapter layer received an Alerting response from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Prack Request</td>
<td>Adapter layer received a Prack request from the protocol stack.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Find Codec in Prack Request Failed</td>
<td>SIP failed to get the codec in the Prack request.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Prack Response</td>
<td>Adapter layer received a response to the Prack request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Update Request</td>
<td>Adapter layer received an Update request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Update Response</td>
<td>Adapter layer received a response to the Update request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Connect Indication</td>
<td>Adapter layer received a 200 OK response from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Connect Ack Request</td>
<td>Adapter layer received an ACK from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Re_Invite Request</td>
<td>Adapter layer received a Re_Invite request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Re_Invite Response</td>
<td>Adapter layer received a response to the Re_Invite request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Re_Invite Ack Request</td>
<td>Adapter layer received a Re_Invite Ack Request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Release Indication</td>
<td>Adapter layer received a Release message from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Release Response</td>
<td>Adapter layer received a response to the Release message from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Release Complete Indication</td>
<td>Adapter layer received an error response to the Invite request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Notify Request</td>
<td>Adapter layer received a Notify request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Notify Response</td>
<td>Adapter layer received a response to the Notify request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Release Complete Sub Indication</td>
<td>Adapter layer received error responses to the REFER and NOTIFY requests from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Information Request</td>
<td>Adapter layer received an Info request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Information Response</td>
<td>Adapter layer received a response to the Info request from the protocol stack.</td>
</tr>
<tr>
<td>Stack --&gt; Adapter : Feature Response</td>
<td>Adapter layer received a response to the Feature request from the protocol stack.</td>
</tr>
<tr>
<td>CMC --&gt; SIP : Accp Setup</td>
<td>CMC sent an ACCP Setup message to SIP.</td>
</tr>
<tr>
<td>CMC --&gt; SIP : XXX</td>
<td>CMC sent a XXX message to SIP.</td>
</tr>
<tr>
<td>SIP Delete Media Channel Success</td>
<td>SIP deleted a media channel successfully.</td>
</tr>
<tr>
<td>SIP ChangeCodec Media Channel Success</td>
<td>SIP changed the codec type for the media channel.</td>
</tr>
<tr>
<td>Configuration OutBandNte ENABLED by FEATURE</td>
<td>Enable Feature to enable the OutBandNte attribute.</td>
</tr>
<tr>
<td>Adapter --&gt; Stack : Setup Ssn Request</td>
<td>Adapter layer sent a Setup request to the protocol stack.</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
Adapter --> Stack : Alerting Request | Adapter layer sent an Alerting request to the protocol stack.
Adapter --> Stack : Connect Request | Adapter layer sent a Connect request to the protocol stack.
Adapter --> Stack : ConnectAck Request | Adapter layer sent a ConnectAck request to the protocol stack.
Adapter --> Stack : Information Request | Adapter layer sent an Info request to the protocol stack.
Adapter --> Stack : Release Response | Adapter layer sent an error response to the Setup message to the protocol stack.
Adapter --> Stack : XX Ssn Request | Adapter layer sent a XX request to the SSN module of the protocol stack.
Adapter --> Stack : XX Ssn Ack Request | Adapter layer sent an ACK for XX request to the SSN module of the protocol stack.
Succeed to send Subscribe_Release Msg | SIP succeeded in sending a Subscribe_Release message to the protocol stack.
Encode SipFrag Success | SIP succeeded in encoding SipFrag message.
Analyze the destination of INVITE request: Current setup type (= %d) cannot be supported. | Analyze the destination of INVITE request: Current setup type (= %d) was not supported.
Start get next SUBSCRIBE redirect server. | SIP started to obtain the next SUBSCRIBE redirect server.
Succeed in getting next redirect IPv4 server. | SIP succeeded in obtaining the next redirect server with an IPv4 address.

### Examples

The output in the following examples was created when User A with number 1000 calls User B with number 3000 using a out-of-band method:

```bash
# Enable debugging for SIP events on the originating device.
<Sysname> debugging voice sip event
   Enable SIP EVENT message debugging functions
<Sysname>
*May 12 11:14:58:514 2007 Sysname SIP/7/VOICE:
  SIP_Event: CMC --> SIP : Accp Setup.
  // CMC sent an ACCP Setup message to SIP.
*May 12 11:14:58:514 2007 Sysname SIP/7/VOICE:
  // SIP replied with an ACCP Setup Ack to CMC.
*May 12 11:14:58:514 2007 Sysname SIP/7/VOICE:
  SIP_Event: Get Nte From DPL : Configuration OutBandNte DISABLE.
  // The OutBandNte attribute was not enabled on the local end.
*May 12 11:14:58:514 2007 Sysname SIP/7/VOICE:
  // The SIP adapter layer sent a Setup request to the protocol stack.
```
*May 12 11:14:58:515 2007 Sysname SIP/7/VOICE:
SIP_Event: Stack --> Adapter : Setup Ack

// The protocol stack sent a Setup Ack to the SIP adapter layer.

*May 12 11:14:58:550 2007 Sysname SIP/7/VOICE:
SIP_Event: Stack --> Adapter : Alerting Indication.

// The protocol stack sent an Alerting message to the SIP adapter layer.

*May 12 11:14:58:550 2007 Sysname SIP/7/VOICE:
SIP_Event: SIP --> CMC : Accp Alerting.

// SIP sent an ACCP Alerting message to CMC.

*May 12 11:14:58:550 2007 Sysname SIP/7/VOICE:
SIP_Event: SIP Get PayLoad Size: Use the Payload Size[ 30 ].

// The packetization period is 30 ms.

*May 12 11:14:58:550 2007 Sysname SIP/7/VOICE:
SIP_Event: SIP --> CMC : Accp ChannelReady.

// SIP sent an ACCP ChannelReady message to CMC.

*May 12 11:14:58:649 2007 Sysname SIP/7/VOICE:
SIP_Event: CMC --> SIP : Accp ChannelReadyAck.

// CMC sent an ACCP ChannelReadyAck message to SIP.

*May 12 11:14:58:699 2007 Sysname SIP/7/VOICE:
SIP_Event: VIM Attach IppId(= 107) to Line(= 2883680) Success.

// Ippid 107 was registered to line 2883680 successfully on VIM.

*May 12 11:14:58:849 2007 Sysname SIP/7/VOICE:
SIP_Event: SIP Create Meida Channel:
  Local Address = 3.1.1.19 : 16598
  Remote Address = 3.1.1.29 : 16560
  Encode Type = 11

// SIP set up a media channel. The IP address of the calling party is 3.1.1.19, the IP address of the called
party is 3.1.1.29, and the codec type is G.729.

*May 12 11:15:03:471 2007 Sysname SIP/7/VOICE:
SIP_Event: Stack --> Adapter : Connect Indication.

// The protocol stack sent a Connect message to the SIP adapter layer.

*May 12 11:15:03:471 2007 Sysname SIP/7/VOICE:
SIP_Event: SIP --> CMC : Accp Connect.

// SIP sent an ACCP Connect message to CMC.

*May 12 11:15:03:471 2007 Sysname SIP/7/VOICE:
SIP_Event: Adapter --> Stack : ConnectAck Request.

// The SIP adapter layer sent a ConnectAck message to the protocol stack.

*May 12 11:15:03:472 2007 Sysname SIP/7/VOICE:
SIP_Event: SIP Get PayLoad Size: Use the Payload Size[ 30 ].

// The packetization period is 30 ms.

*May 12 11:15:03:472 2007 Sysname SIP/7/VOICE:
SIP_Event: SIP --> CMC : Accp Code Switch

// SIP sent an ACCP Code Switch message to CMC to change the codec type.

*May 12 11:15:03:472 2007 Sysname SIP/7/VOICE:
SIP_Event:  SIP ChangeCodec Media Channel: SIP ChangeCodec Media Channel Success!
// SIP changed the codec type for the media channel.

*May 12 11:15:03:473 2007 Sysname SIP/7/VOICE:
SIP_Event:  SIP Delete Media Channel: SIP Delete Media Channel Success!
// SIP deleted a media channel.

*May 12 11:15:03:473 2007 Sysname SIP/7/VOICE:
SIP_Event:  SIP Create Media Channel:
  Local Address = 3.1.1.19 : 16598
  Remote Address = 3.1.1.29 : 16560
  Encode Type    = 11
// SIP set up a media channel. The IP address of the calling party is 3.1.1.19, the IP address of the called
party is 3.1.1.29, and the codec type is G.729.

*May 12 11:15:03:473 2007 Sysname SIP/7/VOICE:
SIP_Event:  SIP --> CMC : Accp Information.
  Enable Outband Sip
// SIP sent an ACCP Information message to CMC to indicate that DTMF detection had been enabled.

*May 12 11:15:25:19 2007 Sysname SIP/7/VOICE:
SIP_Event:  SIP Delete Media Channel: SIP Delete Media Channel Success!
*May 12 11:15:25:20 2007 Sysname SIP/7/VOICE:
SIP_Event:  SIP --> CMC : Accp Information.
  Disable Outband Sip
// SIP sent an ACCP Information message to CMC to indicate that DTMF detection had been disabled.

*May 12 11:15:25:20 2007 Sysname SIP/7/VOICE:
SIP_Event:  SIP Delete Media Channel: SIP Delete Media Channel Success!
// deleted a media channel.

*May 12 11:15:25:20 2007 Sysname SIP/7/VOICE:
SIP_Event:  Adapter --> Stack : Bye Ssn Request
// The SIP adapter layer sent a BYE request to the protocol stack, and the originating device went
on-hook.

*May 12 11:15:25:20 2007 Sysname SIP/7/VOICE:
SIP_Event:  SIP --> CMC : Accp ReleaseComp.
// SIP sent an ACCP ReleaseComp message to CMC as a confirmation to the ACCP Release.

*May 12 11:15:25:21 2007 Sysname SIP/7/VOICE:
// SIP failed to send a Subscribe_Release message.

*May 12 11:15:25:23 2007 Sysname SIP/7/VOICE:
// The SIP protocol stack sent a response to the BYE request to the adapter layer.

# Enable debugging for SIP messages on the originating device.
<Sysname> debugging voice sip message
 Enable SIP message debugging functions
INVITE sip:3000@3.1.1.29:5060 SIP/2.0
Via: SIP/2.0/UDP 3.1.1.19:5060;branch=z9hG4bK9558081c4a9
Call-ID: 178990f081c919d4752cd8e19558081c109@3.1.1.19
From: <sip:1000@3.1.1.19:5060>;tag=9558081c
To: <sip:3000@3.1.1.29:5060>
CSeq: 1 INVITE
Contact: <sip:1000@3.1.1.19:5060>
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 12 May 2007 11:31:51 GMT
Supported: 100rel
Max-Forwards: 70
Content-Length: 230
Content-Type: application/sdp

v=0
o=H3C 1073742116 1073742116 IN IP4 3.1.1.19
s=Sip Call
c=IN IP4 3.1.1.19
t=0 0
m=audio 16602 RTP/AVP 18 8 0 4
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000

// The originating device sent an INVITE request to the terminating device.

SIP/2.0 100 Trying
Via: SIP/2.0/UDP 3.1.1.19:5060;branch=z9hG4bK9558081c4a9
Call-ID: 178990f081c919d4752cd8e19558081c109@3.1.1.19
From: <sip:1000@3.1.1.19:5060>;tag=9558081c
To: <sip:3000@3.1.1.29:5060>
CSeq: 1 INVITE
Content-Length: 0

// The originating device received a 100 (Trying) response from the terminating device.

SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 3.1.1.19:5060;branch=z9hG4bK9558081c4a9
Call-ID: 178990f081c919d4752cd8e19558081c109@3.1.1.19
From: <sip:1000@3.1.1.19:5060>;tag=9558081c
To: <sip:3000@3.1.1.29:5060>;tag=09804351
CSeq: 1 INVITE
Contact: <sip:3000@3.1.1.29:5060>
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 12 May 2007 11:30:14 GMT
Content-Length: 158
Content-Type: application/sdp

v=0
o=H3C 1073742102 1073742102 IN IP4 3.1.1.29
s=Call
m=audio 16564 RTP/AVP 18
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no

// The originating device received a 183 (Session Progress) response from the terminating device.

*May 12 11:31:54:801 2007 Sysname SIP/7/VOICE:
NetWork-->Stack:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 3.1.1.19:5060;branch=z9hG4bK9558081c4a9
Call-ID: 178990f081c919d4752cd8e19558081c109@3.1.1.19
From: <sip:1000@3.1.1.19:5060>;tag=9558081c
To: <sip:3000@3.1.1.29:5060>;tag=09804351
CSeq: 1 INVITE
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 12 May 2007 11:30:17 GMT
Content-Length: 158
Content-Type: application/sdp

v=0
o=H3C 1073742104 1073742104 IN IP4 3.1.1.29
s=Call
m=audio 16564 RTP/AVP 18
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no

// The terminating device went off-hook, and the originating device received a 200 (OK) response from the terminating device.

*May 12 11:31:54:801 2007 Sysname SIP/7/VOICE:
Stack-->NetWork:
ACK sip:3000@3.1.1.29:5060 SIP/2.0
Via: SIP/2.0/UDP 3.1.1.19:5060;branch=z9hG4bK4ce35b0fac2
Call-ID: 178990f081c919d4752cd8e19558081c109@3.1.1.19
From: <sip:1000@3.1.1.19:5060>;tag=9558081c
To: <sip:3000@3.1.1.29:5060>;tag=09804351
CSeq: 1 ACK
Date: Sat, 12 May 2007 11:31:54 GMT
Max-Forwards: 70
The originating device sent an ACK to confirm the reception of the 200 response.

*May 12 11:33:04:949 2007 Sysname SIP/7/VOICE:
Stack--->NetWork:
BYE sip:3000@3.1.1.29:5060 SIP/2.0
Via: SIP/2.0/UDP 3.1.1.19:5060;branch=z9hG4bKd66d95e966e
Call-ID: 178990f081c919d4752cd8e19558081c01093.1.1.19
From: <sip:1000@3.1.1.19:5060>;tag=9558081c
To: <sip:3000@3.1.1.29:5060>;tag=09804351
CSeq: 2 BYE
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 12 May 2007 11:33:04 GMT
Max-Forwards: 70
Content-Length: 0

The originating device went off-hook and sent a BYE request to the terminating device.

*May 12 11:33:04:952 2007 Sysname SIP/7/VOICE:
NetWork--->Stack:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 3.1.1.19:5060;branch=z9hG4bKd66d95e966e
Call-ID: 178990f081c919d4752cd8e19558081c01093.1.1.19
From: <sip:1000@3.1.1.19:5060>;tag=9558081c
To: <sip:3000@3.1.1.29:5060>;tag=09804351
CSeq: 2 BYE
Content-Length: 0

The originating device received a 200 response from the terminating device as a response to the BYE request.

# Enable debugging for SIP timers on the originating device.
<Sysname>debug voice sip timer
    Enable SIP timer debugging functions
<Sysname>

*May 12 18:12:53:820 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 11 ulIndex = 158  ulDuration = 30000

SIP started a timer (ulTimerGroup = 11 ulIndex = 158), with the value set to 30 seconds.

*May 12 18:12:53:820 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 3 ulIndex = 144  ulDuration = 500

*May 12 18:12:53:821 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 4 ulIndex = 144  ulDuration = 32000

*May 12 18:12:53:821 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 16 ulIndex = 13  ulDuration = 600000

*May 12 18:12:53:821 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 11 ulIndex = 158

SIP stopped the timer (ulTimerGroup = 11 ulIndex = 158).

*May 12 18:12:53:823 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 4 ulIndex = 144

*May 12 18:12:53:824 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 3 ulIndex = 144

*May 12 18:12:53:824 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 4 ulIndex = 144 ulDuration = 64000

*May 12 18:12:53:858 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 4 ulIndex = 144

*May 12 18:12:53:954 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 4 ulIndex = 144 ulDuration = 256000

*May 12 18:12:54:104 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 12 ulIndex = 13 ulDuration = 120000

*May 12 18:12:54:204 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 16 ulIndex = 13

*May 12 18:12:54:354 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 16 ulIndex = 13 ulDuration = 600000

*May 12 18:12:54:454 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 22 ulIndex = 13 ulDuration = 5000

*May 12 18:12:58:957 2007 Sysname SIP/7/VOICE:
SIP_Timer: Timer out ulTimerGroup = 22 ulIndex = 13

// The timer (ulTimerGroup = 22 ulIndex = 13) timed out.

*May 12 18:13:00:640 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 12 ulIndex = 13

*May 12 18:13:00:641 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 12 ulIndex = 13 ulDuration = 64000

*May 12 18:13:00:641 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 16 ulIndex = 13

*May 12 18:13:00:641 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 16 ulIndex = 13 ulDuration = 600000

*May 12 18:13:00:642 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 11 ulIndex = 159 ulDuration = 30000

*May 12 18:13:00:642 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 12 ulIndex = 13

*May 12 18:13:00:642 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 16 ulIndex = 13

*May 12 18:13:00:643 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 16 ulIndex = 13  ulDuration = 1800000

*May 12 18:13:00:643 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 22 ulIndex = 13

*May 12 18:13:00:786 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 22 ulIndex = 13  ulDuration = 5000

*May 12 18:13:00:896 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 4 ulIndex = 144

*May 12 18:13:00:996 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 11 ulIndex = 159

*May 12 18:13:01:146 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 11 ulIndex = 160  ulDuration = 32000

*May 12 18:13:05:659 2007 Sysname SIP/7/VOICE:
SIP_Timer: Timer out ulTimerGroup = 22 ulIndex = 13

*May 12 18:13:06:490 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 22 ulIndex = 13

*May 12 18:13:06:491 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 11 ulIndex = 160  ulDuration = 30000

*May 12 18:13:06:491 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 3 ulIndex = 145  ulDuration = 500

*May 12 18:13:06:491 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 4 ulIndex = 145  ulDuration = 32000

*May 12 18:13:06:492 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 12 ulIndex = 13

*May 12 18:13:06:492 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 16 ulIndex = 13

*May 12 18:13:06:492 2007 Sysname SIP/7/VOICE:
SIP_Timer: Start Timer ulTimerGroup = 16 ulIndex = 13  ulDuration = 600000

*May 12 18:13:06:493 2007 Sysname SIP/7/VOICE:
SIP_Timer: Stop Timer ulTimerGroup = 11 ulIndex = 160

*May 12 18:13:06:495 2007 Sysname SIP/7/VOICE:
debugging voice ssm call

Use `debugging voice ssm call` to enable debugging for SIP server basic call functions.
Use `undo debugging voice ssm call` to disable debugging for SIP server basic call functions.

**Syntax**

```
debugging voice ssm call { all | error | event | fsm | info | prim | timer }
undo debugging voice ssm call { all | error | event | fsm | info | prim | timer }
```

**Default**

Debugging for SIP server basic call functions is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **call**: Enables debugging for SIP server basic call functions.
- **all**: Specifies all types of debugging.
- **error**: Specifies error debugging.
- **event**: Specifies event debugging.
- **fsm**: Specifies finite state machine debugging.
- **info**: Specifies information debugging.
- **prim**: Specifies primitive debugging.
- **timer**: Specifies timer debugging.

**Examples**

The output in the following examples was created when User A with number 2002 as the calling party establishes a call with User B with number 200:

# Enable all types of debugging for SIP server basic call functions.
```bash
<Sysname> debugging voice ssm call all
```
Enable SSM CALL ALL debugging functions
<Sysname>

*Sep 7 16:38:37:637 2007 Sysname SSM/7/VOICE:
  SSM_INFO [CALL_SSA]:
  General: Succeed in getting a free element from the list.
  Additional: Call control block ID is 42.

// The SSA module got a free element from the list as the control block (ID: 42) for the current call.

*Sep 7 16:38:37:638 2007 Sysname SSM/7/VOICE:
  SSM_INFO [CALL_SSA]:
  General: Received message from the stack.
  Additional: The message type is SetupInd.
  Call control block ID is 42.
  Caller number is 2002.
  Called number is 200.

// The SSA module received a SetupInd message from the network side.

*Sep 7 16:38:37:638 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SSA]:
  Send UCM Setup request message.

// The SSA module sent a UCM_SETUPREQ message to the SLC module.

*Sep 7 16:38:37:638 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SLC]:
  Receive UCM Setup request message.

// The SLC module received the UCM_SETUPREQ message.

*Sep 7 16:38:37:638 2007 Sysname SSM/7/VOICE:
  SSM_INFO [CALL_SLC]:
  General: Succeed in getting a free element from the list.
  Additional: Conference control block ID is 15.

// The SLC module got a free element from the list as the conference control block (ID: 15).

*Sep 7 16:38:37:638 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [CALL_SLC]:
  CALLCTRL: Received event CALLESTABLISH while at state INIT.(ObjectID = 15)
// The state machine CALLCTRL received a CALLESTABLISH event in the INIT state.

*Sep 7 16:38:37:638 2007 Sysname SSM/7/VOICE:
  SSM_INFO [CALL_SLC]:
  General: The authentication is needed.

// Authentication is needed for this call.

*Sep 7 16:38:37:638 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SLC]:
  Send UCM Setup reject message.

// The SLC module sent a UCM_SETUPREJ message to the SSA module.

*Sep 7 16:38:37:639 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
  Start Timer: Type = 4, Group = 4, Index = 15, Length = 8000
// The SLC module started a timer, with the value set to 8000 ms.

*Sep 7 16:38:37:840 2007 Sysname SSM/7/VOICE:
SSM_FSM [CALL_SLC]:
CALLCTRL: State changed from INIT to CALLINPROGRESS.(ObjectID = 15)
// The state machine CALLCTRL transitioned from INIT to CALLINPROGRESS.

*Sep 7 16:38:38:102 2007 Sysname SSM/7/VOICE:
SSM_PRIM [CALL_SSA]:
Receive UCM Setup reject message.
// The SSA module received the UCM_SETUPREJ message.

*Sep 7 16:38:38:253 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SSA]:
General: Succeed in sending the network message.
Additional: Method type is INVITE, status code is 407.
// The SSA module sent a 407 response to the network side for the INVITE method.

*Sep 7 16:38:38:354 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SSA]:
General: Succeed in freeing element to the list.
Additional: Call control block ID is 42.
// The SSA module released the current call control block (ID: 42).

*Sep 7 16:38:38:556 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SSA]:
General: Succeed in getting a free element from the list.
Additional: Call control block ID is 43.
// The SSA module got a free element from the list as the control block (ID: 43) for the current call.

*Sep 7 16:38:38:707 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SSA]:
General: Received message from the stack.
Additional: The message type is SetupInd.
Call control block ID is 43.
Caller number is 2002.
Called number is 200.
// The SSA module received a SetupInd message from the network side.

*Sep 7 16:38:38:858 2007 Sysname SSM/7/VOICE:
SSM_PRIM [CALL_SSA]:
Send UCM Setup request message.
// The SSA module sent a UCM_SETUPREQ message to the SLC module.

*Sep 7 16:38:39:959 2007 Sysname SSM/7/VOICE:
SSM_PRIM [CALL_SLC]:
Receive UCM Setup request message.
// The SLC module received the UCM_SETUPREQ message.

*Sep 7 16:38:39:110 2007 Sysname SSM/7/VOICE:
SSM_EVENT [CALL_SLC]:
CALLCTRL: Received event CALLESTABLISH while at state CALLINPROGRESS.(ObjectID = 15)
// The state machine CALLCTRL received a CALLESTABLISH event in the CALLINPROGRESS state.

*Sep 7 16:38:39:261 2007 Sysname SSM/7/VOICE:
SSM_TIMER [CALL_SLC]:
Stop Timer: Type = 4, Group = 4, Index = 15, Length = 8000
// The SLC module stopped an 8000-ms timer.

*Sep 7 16:38:39:362 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SLC]:
  General: Succeed in getting a free element from the list.
  Additional: Call control block ID is 22.

// The SSA module got a free element from the list as the call control block (ID: 22).

*Sep 7 16:38:39:573 2007 Sysname SSM/7/VOICE:
SSM_EVENT [CALL_SLC]:
  CALLSRV: Received event UCMSETUPREQ while at state IDLE.(ObjectID = 22)

// The state machine CALLSRV received a UCMSETUPREQ event in the IDLE state.

*Sep 7 16:38:39:775 2007 Sysname SSM/7/VOICE:
SSM_PRIM [CALL_SLC]:
  Send UCM Setup ACK message.

// The SLC module sent a UCM_SETUACK message to the SSA module.

*Sep 7 16:38:39:926 2007 Sysname SSM/7/VOICE:
SSM_TIMER [CALL_SLC]:
  Start Timer: Type = 6, Group = 6, Index = 22, Length = 16000

// The SLC module started a timer, with the value set to 16000 ms.

*Sep 7 16:38:40:228 2007 Sysname SSM/7/VOICE:
SSM_FSM [CALL_SLC]:
  CALLSRV: State changed from IDLE to SETUP.(ObjectID = 22)

// The state machine CALLSRV transitioned from IDLE to SETUP.

*Sep 7 16:38:40:329 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SLC]:
  General: Succeed in getting a free element from the list.
  Additional: Call control block ID is 23.

// The SLC module got a free element from the list as the call control block (ID: 23).

*Sep 7 16:38:40:430 2007 Sysname SSM/7/VOICE:
SSM_EVENT [CALL_SLC]:
  CALLCLT: Received event CCSETUPREQ while at state IDLE.(ObjectID = 23)

// The state machine CALLCLT received a CCSETUPREQ event in the IDLE state.

*Sep 7 16:38:40:631 2007 Sysname SSM/7/VOICE:
SSM_PRIM [CALL_SLC]:
  Send UCM Setup request message.

// The SLC module sent a UCM_SETUPREQ message to the SSA module.

*Sep 7 16:38:40:782 2007 Sysname SSM/7/VOICE:
SSM_TIMER [CALL_SLC]:
Start Timer: Type = 6, Group = 6, Index = 23, Length = 16000

// The SLC module started a timer, with the value set to 16000 ms.

*Sep 7 16:38:40:934 2007 Sysname SSM/7/VOICE:
SSM_FSM [CALL_SLC]:
CALLCLT: State changed from IDLE to SETUP.(ObjectID = 23)

// The state machine CALLCLT transitioned from IDLE to SETUP.

*Sep 7 16:38:41:135 2007 Sysname SSM/7/VOICE:
SSM_FSM [CALL_SLC]:
CALLCTRL: State changed from CALLINPROGRESS to INCALL.(ObjectID = 15)

// The state machine CALLCTRL transitioned from CALLINPROGRESS to INCALL.

*Sep 7 16:38:41:246 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SLC]:
General: Succeed in dealing with call request.
Additional: Conference ID is 15.
Incoming call ID is 22.
Outgoing call ID is 23.
Caller number is 2002.
Called number is 200.

// The SLC module succeeded in handling the call request.

*Sep 7 16:38:41:397 2007 Sysname SSM/7/VOICE:
SSM_PRIM [CALL_SSA]:
Receive UCM Setup ACK message.

// The SSA module received the UCM_SETU PACK message.

*Sep 7 16:38:41:498 2007 Sysname SSM/7/VOICE:
SSM_PRIM [CALL_SSA]:
Receive UCM Setup request message.

// The SSA module received the UCM_SETUPREQ message.

*Sep 7 16:38:41:699 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SSA]:
General: Succeed in getting a free element from the list.
Additional: Call control block ID is 44.

// The SSA module got a free element from the list as the control block (ID: 44) for the current call.

*Sep 7 16:38:41:851 2007 Sysname SSM/7/VOICE:
SSM_INFO [CALL_SSA]:
General: Received message from the stack.
Additional: The message type is SetupAck.

// The SSA module received a SetupAck message from the network side.
// The SSA module sent a UCM_SETTACK message to the SLC module.

// The SSA module sent an INVITE to the network side.

// The SLC module received the UCM_SETTACK message.

// The state machine CALLCLT received a UCM_SETTACK event in the SETUP state.

// The SSA module received an AlertingInd message from the network side.

// The SSA module sent a UCM_ALERTING message to the SLC module.

// The SLC module received the UCM_ALERTING message.

// The state machine CALLCLT received a UCMAERTING event in the SETUP state.
CALLCTRL: Received event CALLTRANSMIT while at state INCALL. (ObjectID = 15)

// The state machine CALLCTRL received a CALLTRANSMIT event in the INCALL state.

*Sep 7 16:38:43 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [CALL_SLC]:
    CALLSRV: Received event CCALERTINGIND while at state SETUP. (ObjectID = 22)

// The state machine CALLSRV received a CCALERTING event in the SETUP state.

*Sep 7 16:38:43:624 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SLC]:
    Send UCM Alerting message.

// The SLC module sent a UCM_ALERTING message to the SSA module.

*Sep 7 16:38:43:725 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
    Stop Timer: Type = 6, Group = 6, Index = 22, Length = 16000

// The SLC module stopped a 16000-ms timer.

*Sep 7 16:38:43:876 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
    Start Timer: Type = 6, Group = 6, Index = 22, Length = 150000

// The SLC module started a timer, with the value set to 150000 ms.

*Sep 7 16:38:44:77 2007 Sysname SSM/7/VOICE:
  SSM_FSM [CALL_SLC]:
    CALLSRV: State changed from SETUP to ALERTING. (ObjectID = 22)

// The state machine CALLSRV transitioned from SETUP to ALERTING.

*Sep 7 16:38:44:178 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
    Stop Timer: Type = 6, Group = 6, Index = 23, Length = 16000

// The SLC module stopped a 16000-ms timer.

*Sep 7 16:38:44:289 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
    Start Timer: Type = 6, Group = 6, Index = 23, Length = 150000

// The SLC module started a timer, with the value set to 150000 ms.

*Sep 7 16:38:44:440 2007 Sysname SSM/7/VOICE:
  SSM_FSM [CALL_SLC]:
    CALLCLT: State changed from SETUP to ALERTING. (ObjectID = 23)

// The state machine CALLCLT transitioned from SETUP to ALERTING.

*Sep 7 16:38:44:642 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SSA]:
Receive UCM Alerting message.

// The SSA module received a UCM_ALERTING message.

*Sep  7 16:38:44:793 2007 Sysname SSM/7/VOICE:
  SSM_INFO [CALL_SSA]:
  General: Succeed in sending the network message.
  Additional: Method type is INVITE, status code is 183.

// The SSA module sent a 183 response to the network side for the INVITE method.

*Sep  7 16:38:45:50 2007 Sysname SSM/7/VOICE:
  SSM_INFO [CALL_SSA]:
  General: Received message from the stack.
  Additional: The message type is ConnectInd.

// The SSA module received a ConnectInd message from the network side.

*Sep  7 16:38:45:50 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SSA]:
  Send UCM Connecting message.

// The SSA module sent a UCM_CONNECTING message to the SLC module.

*Sep  7 16:38:45:196 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SLC]:
  Receive UCM Connecting message.

// The SLC module received the UCM_CONNECTING message.

*Sep  7 16:38:45:347 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [CALL_SLC]:
  CALLCLT: Received event UCMCONNECTING while at state ALERTING.(ObjectID = 23)

// The state machine CALLCLT received a UCMCONNECTING event in the ALERTING state.

*Sep  7 16:38:45:498 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [CALL_SLC]:
  CALLCTRL: Received event CALLTRANSMIT while at state INCALL.(ObjectID = 15)

// The state machine CALLCTRL received a CALLTRANSMIT event in the INCALL state.

*Sep  7 16:38:45:649 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [CALL_SLC]:
  CALLSRV: Received event CCCONNECTIND while at state ALERTING.(ObjectID = 22)

// The state machine CALLSRV received a CCCONNECTING event in the ALERTING state.

*Sep  7 16:38:45:800 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SLC]:
  Send UCM Connecting message.

// The SLC module sent a Connecting message to the SSA module.
*Sep  7 16:38:45:901 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
  Stop Timer: Type = 6, Group = 6, Index = 22, Length = 150000
  // The SLC module stopped a 150000-ms timer.

*Sep  7 16:38:46:103 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
  Start Timer: Type = 6, Group = 6, Index = 22, Length = 16000
  // The SLC module started a timer, with the value set to 16000 ms.

*Sep  7 16:38:46:254 2007 Sysname SSM/7/VOICE:
  SSM_FSM [CALL_SLC]:
  CALLSRV: State changed from ALERTING to CONNECT.(ObjectID = 22)
  // The state machine CALLSRV transitioned from ALERTING to CONNECT.

*Sep  7 16:38:46:415 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
  Stop Timer: Type = 6, Group = 6, Index = 23, Length = 150000
  // The SLC module stopped a 150000-ms timer.

*Sep  7 16:38:46:516 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
  Start Timer: Type = 6, Group = 6, Index = 23, Length = 16000
  // The SLC module started a timer, with the value set to 16000 ms.

*Sep  7 16:38:46:667 2007 Sysname SSM/7/VOICE:
  SSM_FSM [CALL_SLC]:
  CALLCLT: State changed from ALERTING to CONNECT.(ObjectID = 23)
  // The state machine CALLCLT transitioned from ALERTING to CONNECT.

*Sep  7 16:38:46:818 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SSA]:
  Receive UCM Connecting message.
  // The SSA module received a UCM_CONNECTING message.

*Sep  7 16:38:47:20 2007 Sysname SSM/7/VOICE:
  SSM_INFO [CALL_SSA]:
  General: Succeed in sending the network message.
  Additional: Method type is INVITE, status code is 200.
  // The SSA module sent a 200 response to the network side for the INVITE method.

*Sep  7 16:38:47:121 2007 Sysname SSM/7/VOICE:
  SSM_INFO [CALL_SSA]:
  General: Received message from the stack.
  Additional: The message type is ConnectCfm.
// The SSA module received a ConnectCfm message from the network side.

*Sep  7 16:38:47:272 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SSA]:
  Send UCM Connect ACK message.

// The SSA module sent a UCM_CONNECTACK message to the SLC module.

*Sep  7 16:38:47:423 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [CALL_SLC]:
  Receive UCM Connect ACK message.

// The SLC module received the UCM_CONNECTACK message.

*Sep  7 16:38:47:574 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [CALL_SLC]:
  CALLSRV: Received event UCMCONNECTCFM while at state CONNECT.(ObjectID = 22)

// The state machine CALLSRV received a UCMCONNECTCFM event in the CONNECT state.

*Sep  7 16:38:47:725 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [CALL_SLC]:
  CALLCTRL: Received event CALLCONFIRM while at state INCALL.(ObjectID = 15)

// The state machine CALLCTRL received a CALLCONFIRM event in the INCALL state.

*Sep  7 16:38:47:886 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [CALL_SLC]:
  CALLCLT: Received event CCCONNECTCFM while at state CONNECT.(ObjectID = 23)

// The state machine CALLCLT received a CCCONNECTCFM event in the CONNECT state.

*Sep  7 16:38:48:37 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
  Start Timer: Type = 6, Group = 6, Index = 23, Length = 16000

// The SLC module sent a UCM_CONNECTACK message to the SSA module.

*Sep  7 16:38:48:138 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
  Stop Timer: Type = 6, Group = 6, Index = 23, Length = 16000

// The SLC module stopped a 16000-ms timer.

*Sep  7 16:38:48:340 2007 Sysname SSM/7/VOICE:
  SSM_FSM [CALL_SLC]:
  CALLCLT: State changed from CONNECT to ACTIVE.(ObjectID = 23)

// The state machine CALLCLT transitioned from CONNECT to ACTIVE.

*Sep  7 16:38:48:491 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [CALL_SLC]:
  Start Timer: Type = 5, Group = 5, Index = 15, Length = 3600000
// The SLC module started a timer, with the value set to 3600000 ms.

*Sep  7 16:38:48:642 2007 Sysname SSM/7/VOICE:
    SSM_TIMER [CALL_SLC]:
        Stop Timer: Type = 6, Group = 6, Index = 22, Length = 16000

// The SLC module stopped a 16000-ms timer.

*Sep  7 16:38:48:743 2007 Sysname SSM/7/VOICE:
    SSM_TIMER [CALL_SLC]:
        CALLSRV: State changed from CONNECT to ACTIVE.(ObjectID = 22)

// The state machine CALLSRV transitioned from CONNECT to ACTIVE.

*Sep  7 16:38:48:894 2007 Sysname SSM/7/VOICE:
    SSM_PRIM [CALL_SSA]:
        Receive UCM Connect ACK message.

// The SSA module received a UCM_CONNECTACK message.

*Sep  7 16:38:49:976 2007 Sysname SSM/7/VOICE:
    SSM_INFO [CALL_SSA]:
        General: Succeed in sending the network message.
        Additional: Method type is ACK.

// The SSA module sent an ACK to the network side.

debugging voice ssm lsm

Use debugging voice ssm lsm to enable debugging for SIP server location service.
Use undo debugging voice ssm lsm to disable debugging for SIP server location service.

Syntax

d debugging voice ssm lsm { all | error | info }
undo debugging voice ssm lsm { all | error | info }

Default

Debugging for SIP server location service is disabled.

Views

User view

Default command level

2: System level

Parameters

lsm: Enables debugging for SIP server location service.
all: Specifies all types of debugging.
error: Specifies error debugging.
info: Specifies information debugging.
Examples

# Enable all types of debugging for location service. When a user with number 2002 initiates a registration request with authentication information to the SIP server, output similar to the following example is generated:

```
<Sysname> debugging voice ssm lsm all
    Enable SSM LSM ALL debugging functions
<Sysname>
```

```
*Sep  8 15:22:09:239 2007 Sysname SSM/7/VOICE:
    SSM_INFO [LSM]:
    General: Succeed in getting the user information.
    Additional: The number is 2002.

    // The location service succeeded in getting user information.
```

```
*Sep  8 15:22:09:240 2007 Sysname SSM/7/VOICE:
    SSM_INFO [LSM]:
    General: The authentication is needed.
    Additional: The number is 2002, the realm is "H3C", the algorithm is "MD5", the nonce is "46e2bf4d2871e5e985f96008c04a0f7c5a85f106".

    // Authentication is needed.
```

```
*Sep  8 15:22:09:240 2007 Sysname SSM/7/VOICE:
    SSM_INFO [LSM]:
    General: Don't pass the authentication successfully.
    Additional: The number is 2002.

    // The authentication failed.
```

```
*Sep  8 15:22:09:243 2007 Sysname SSM/7/VOICE:
    SSM_INFO [LSM]:
    General: Succeed in getting the user information.
    Additional: The number is 2002.

    // The location service succeeded in getting user information.
```

```
*Sep  8 15:22:09:243 2007 Sysname SSM/7/VOICE:
    SSM_INFO [LSM]:
    General: Succeed in passing the authentication.
    Additional: The number is 2002.

    // The authentication succeeded.
```

debugging voice ssm probe

Use **debugging voice ssm probe** to enable debugging for SIP server probe.

Use **undo debugging voice ssm probe** to disable debugging for SIP server probe.

Syntax

```
debugging voice ssm probe { all | error | event | fsm | info | prim | timer }
undo debugging voice ssm probe { all | error | event | fsm | info | prim | timer }
```
**Default**

Debugging for SIP server probe is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- `probe`: Enables debugging for SIP server probe.
- `all`: Specifies all types of debugging.
- `error`: Specifies error debugging.
- `event`: Specifies event debugging.
- `fsm`: Specifies finite state machine debugging.
- `info`: Specifies information debugging.
- `prim`: Specifies primitive debugging.
- `timer`: Specifies timer debugging.

**Examples**

# Enable all types of debugging for SIP server probe. When the server operating in probe mode is disconnected from the primary server, output similar to the following example is generated:

```plaintext
<Sysname> debugging voice ssm lsm all
Enable SSM LSM ALL debugging functions
<Sysname>
*Sep 8 17:44:18:530 2007 Sysname SSM/7/VOICE:
  SSM_EVENT [PROBE_SLC]:
  PROBE: Received event STARTALIVEPROBE while at state IDLE.(ObjectID = 0)
// The Probe state machine received a STARTALIVEPROBE event in the IDLE state.

*Sep 8 17:44:18:530 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [PROBE_SLC]:
  Start Timer: Type = 1, Group = 1, Index = 0, Length = 64000
// The SLC module started a probe timer, with the value set to 64000 ms.

*Sep 8 17:44:18:530 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [PROBE_SLC]:
  Send UCM Options request message.
// The SLC module sent a UCM_OPTIONSREQ message to the SSA module.

*Sep 8 17:44:18:530 2007 Sysname SSM/7/VOICE:
  SSM_FSM [PROBE_SLC]:
  PROBE: State changed from IDLE to ACTIVE.(ObjectID = 0)
// The Probe state machine transitioned from IDLE to ACTIVE.
```
SSM_PRIM [PROBE_SSA]:
Receive UCM Options request message.

// The SSA module received the UCM_OPTIONSREQ message.

SSM_INFO [PROBE_SSA]:
General: Succeed in sending the network message.
Additional: Method type is OPTIONS.

// The SSA module sent an OPTIONS request to the network side.

SSM_PRIM [PROBE_SLC]:
Send UCM Options response message.

// The SSA module sent a UCM_OPTIONSRSP message to the SLC module.

SSM_INFO [PROBE_SLC]:
General: Received message from the stack.
Additional: The message type is options response, response code is 408.

// The SSA module received an OPTIONS response (408) from the protocol stack.

SSM_PRIM [PROBE_SLC]:
Receive UCM Options response message.

// The SLC module received the UCM_OPTIONSRSP message.

SSM_EVENT [PROBE_SLC]:
PROBE: Received event UCMOPTIONCFM while at state ACTIVE.(ObjectID = 0)

// The Probe state machine received a UCMOPTIONCFM event in the ACTIVE state.

SSM_INFO [PROBE_SLC]:
General: Receive options response message.
Additional: Response code is 408.

// The SLC module received the OPTIONS response (408).

SSM_INFO [PROBE_SLC]:
General: Link state change to disconnect.

// The link state changed to disconnect.
Stop Timer: Type = 1, Group = 1, Index = 0, Length = 64000

// The SLC module stopped a 64000-ms probe timer.

*Sep 8 17:44:33:96 2007 Sysname SSM/7/VOICE:
   SSM_TIMER [PROBE_SLC]:
   Start Timer: Type = 1, Group = 1, Index = 0, Length = 64000

// The SLC module started a probe timer, with the value set to 64000 ms.

debugging voice ssm reg

Use debugging voice ssm reg to enable debugging for SIP server registration service.
Use undo debugging voice ssm reg to disable debugging for SIP server registration service.

Syntax

debugging voice ssm reg { all | error | event | fsm | info | prim | timer }
undo debugging voice ssm reg { all | error | event | fsm | info | prim | timer }

Default

Debugging for SIP server registration service is disabled.

Views

User view

Default command level

2: System level

Parameters

reg: Enables debugging for SIP server registration service.
all: Specifies all types of debugging.
error: Specifies error debugging.
event: Specifies event debugging.
fsm: Specifies finite state machine debugging.
info: Specifies information debugging.
prim: Specifies primitive debugging.
timer: Specifies timer debugging.

Examples

# Enable all types of debugging for SIP server registration service. When a user with number 3001
registers with the SIP server, output similar to the following example is generated:
<Sysname> debugging voice ssm reg all
   Enable SSM REG ALL debugging functions
<Sysname>
*Sep 7 16:51:40:326 2007 Sysname SSM/7/VOICE:
   SSM_INFO [REG_SSA]:
   General: Received message from the stack.
   Additional: The message type is RegisterInd.
   Register number is 3001.
// The SSA module received a REGISTER request from the protocol stack.

*Sep 7 16:51:40:326 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [REG_SSA]:
  Send UCM Register request message.

// The SSA module sent the REGISTER request to the UCM module.

*Sep 7 16:51:40:327 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [REG_SLC]:
  Receive UCM Register request message.

// The SLC module received the REGISTER request from the UCM module.

*Sep 7 16:51:40:327 2007 Sysname SSM/7/VOICE:
  SSM_INFO [REG_SLC]:
  General: Succeed in getting a free element from the list.
  Additional: Register control block ID is 33.

// The SLC module got a free element from the list.

*Sep 7 16:51:40:327 2007 Sysname SSM/7/VOICE:
  SSM_OPERATION [REG_SLC]:
  REGISTER: Received event UCMREGISTERREQ while at state IDLE.(ObjectID = 33)

// The SLC module received a UCMREGISTERREQ event in the IDLE state.

*Sep 7 16:51:40:328 2007 Sysname SSM/7/VOICE:
  SSM_FSM [REG_SLC]:
  REGISTER: State changed from IDLE to REGISTERING.(ObjectID = 33)

// The SLC module changed from IDLE state to REGISTERING state.

*Sep 7 16:51:40:328 2007 Sysname SSM/7/VOICE:
  SSM_INFO [REG_SLC]:
  General: Succeed in dealing with register request.
  Additional: Status code is 401.
  expires is 3600(s).
  user number is 3001.

// The SLC module handled the registration request successfully.

*Sep 7 16:51:40:328 2007 Sysname SSM/7/VOICE:
  SSM_PRIM [REG_SLC]:
  Send UCM Register response message.

// The SLC module sent a REGISTER response to the UCM module.

*Sep 7 16:51:40:328 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [REG_SLC]:
  Start Timer: Type = 2, Group = 2, Index = 33, Length = 64000

// The SLC module started a timer, with the value set to 64000 ms.
The SSA module received the REGISTER response from the UCM module.

The SSA module sent a REGISTER response (401) to the network side.

The SLC module stopped a 64000-ms timer.
debugging voice ssm stack

Use **debugging voice ssm stack** to enable debugging for SIP server protocol stack.

Use **undo debugging voice ssm stack** to disable debugging for SIP server protocol stack.

**Syntax**

```plaintext
debugging voice ssm stack { all | info | packet | timer }
d undo debugging voice ssm stack { all | info | packet | timer }
```

**Default**

Debugging for SIP server protocol stack is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **stack**: Enables debugging for SIP server protocol stack.
- **all**: Specifies all types of debugging.
- **info**: Specifies information debugging.
- **packet**: Specifies packet debugging.
- **timer**: Specifies timer debugging.

**Examples**

The output in the following examples was created when User A with number 2001 calls User B with number 3001. User A and User B have both registered with the SIP server.

# Enable timer debugging for SIP server protocol stack.

```plaintext
<Sysname>debugging voice ssm stack timer
 Enable SSM STACK TIMER debugging functions
<Sysname>
```
*Sep  8 17:50:24:554 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 2550, Group = 10, Index = 74, Length = 64000

// The protocol stack started a timer, with the value set to 64000 ms.

*Sep  8 17:50:24:555 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 0, Group = 8, Index = 127, Length = 30000

*Sep  8 17:50:24:555 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 2805, Group = 15, Index = 21, Length = 600000

*Sep  8 17:50:24:555 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Stop Timer: Type = 0, Group = 8, Index = 127, Length = 30000

// The protocol stack stopped a 30000-ms timer.

*Sep  8 17:50:24:555 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 0, Group = 8, Index = 128, Length = 30000

*Sep  8 17:50:24:556 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 0, Group = 9, Index = 75, Length = 500

*Sep  8 17:50:24:556 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 255, Group = 10, Index = 75, Length = 32000

*Sep  8 17:50:24:556 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 2805, Group = 15, Index = 22, Length = 600000

*Sep  8 17:50:24:557 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Stop Timer: Type = 0, Group = 8, Index = 128, Length = 30000

*Sep  8 17:50:24:901 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Stop Timer: Type = 255, Group = 10, Index = 75, Length = 32000

*Sep  8 17:50:24:901 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Stop Timer: Type = 0, Group = 9, Index = 75, Length = 500

*Sep  8 17:50:24:999 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:

371
Start Timer: Type = 2805, Group = 10, Index = 75, Length = 256000

*Sep  8 17:50:25:149 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Start Timer: Type = 0, Group = 11, Index = 22, Length = 128000

*Sep  8 17:50:25:303 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Stop Timer: Type = 2805, Group = 15, Index = 22, Length = 600000

*Sep  8 17:50:25:410 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Start Timer: Type = 2805, Group = 15, Index = 22, Length = 600000

*Sep  8 17:50:25:774 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Start Timer: Type = 2805, Group = 10, Index = 75, Length = 256000

*Sep  8 17:50:25:774 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Stop Timer: Type = 2805, Group = 10, Index = 75, Length = 256000

*Sep  8 17:50:26:340 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Start Timer: Type = 2805, Group = 15, Index = 22, Length = 600000

*Sep  8 17:50:26:340 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Stop Timer: Type = 2805, Group = 15, Index = 22, Length = 600000

*Sep  8 17:50:26:499 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Stop Timer: Type = 2550, Group = 10, Index = 74, Length = 64000

*Sep  8 17:50:26:638 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:
  Start Timer: Type = 2550, Group = 10, Index = 74, Length = 128000

*Sep  8 17:50:26:769 2007 Sysname SSM/7/VOICE:
  SSM_TIMER [STACK]:

372
Start Timer: Type = 0, Group = 8, Index = 129, Length = 30000

*Sep  8 17:50:26:899 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Stop Timer: Type = 2805, Group = 15, Index = 21, Length = 600000

*Sep  8 17:50:27:20 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 2805, Group = 15, Index = 21, Length = 600000

*Sep  8 17:50:27:132 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 1020, Group = 12, Index = 21, Length = 120000

*Sep  8 17:50:27:355 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Stop Timer: Type = 2550, Group = 10, Index = 74, Length = 128000

*Sep  8 17:50:27:464 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 2550, Group = 10, Index = 74, Length = 128000

*Sep  8 17:50:27:713 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Start Timer: Type = 0, Group = 8, Index = 130, Length = 30000

*Sep  8 17:50:27:714 2007 Sysname SSM/7/VOICE:
SSM_TIMER [STACK]:
Stop Timer: Type = 1020, Group = 12, Index = 21, Length = 120000

# Enable packet debugging for SIP server protocol stack.
<Sysname>/debug voice ssm stack packet
Enable SSM STACK PACKET debugging functions
<Sysname>

*Sep  8 17:40:32:561 2007 Sysname SSM/7/VOICE:
SSM_PACKET [STACK]: NetWork--->Stack

INVITE sip:3001@100.1.1.188:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.101:5060;branch=z9hG4bK275cfa6341
Call-ID: b824639c89ea2b98bc6341
From: <sip:2001@192.168.1.101:5060>;tag=275cfa6
To: <sip:3001@100.1.1.188:5060>
CSeq: 1 INVITE
Contact: <sip:2001@192.168.1.101:5060>
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 08 Sep 2007 17:40:32 GMT
Supported: 100rel
Max-Forwards: 70
Content-Length: 239
Content-Type: application/sdp

v=0
o=HUAWEI 1073741844 1073741844 IN IP4 100.1.1.188
s=Sip Call
c=IN IP4 100.1.1.188
t=0 0
m=audio 16396 RTP/AVP 18 8 0 4
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000

// The server received an INVITE request from the network side.

*Sep 8 17:40:32:562 2007 Sysname SSM/7/VOICE:
  SSM_PACKET [STACK]: Stack--->NetWork

SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.101:5060;branch=z9hG4bK275cffa6341
Call-ID: b824639c89ea2bdb3fc82faf275cffe660192.168.1.101
From: <sip:2001@192.168.1.101:5060>;tag=275cffe6
To: <sip:3001@100.1.1.188:5060>
CSeq: 1 INVITE
Content-Length: 0

// The server sent a 100 (Trying) response to the network side.

*Sep 8 17:40:32:562 2007 Sysname SSM/7/VOICE:
  SSM_PACKET [STACK]: Stack--->NetWork

INVITE sip:3001@100.1.1.185:5060;user=phone;ttl=0 SIP/2.0
Via: SIP/2.0/UDP 100.1.1.188:5060;branch=z9hG4bK423ec20d0a4
Call-ID: 3411a7fdeb824639cd797366b423ec20d@100.1.1.188
From: <sip:2001@100.1.1.188:5060;user=phone;ttl=0>;tag=423ec20d
To: <sip:3001@100.1.1.185:5060;user=phone;ttl=0>
CSeq: 1 INVITE
Contact: <sip:Sip-Server@100.1.1.188:5060;user=phone;ttl=0>
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 08 Sep 2007 17:40:32 GMT
Max-Forwards: 70
Supported: 100rel
Content-Length: 239
Content-Type: application/sdp

v=0
o=HUAWEI 1073741844 1073741844 IN IP4 100.1.1.188
s=Sip Call
c=IN IP4 100.1.1.188
t=0 0
m=audio 16396 RTP/AVP 18 8 0 4
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000

// The server sent an INVITE to the network side.

*Sep 8 17:40:32:622 2007 Sysname SSM/7/VOICE:
  SSM_PACKET [STACK]: NetWork-->Stack

SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 100.1.1.188:5060;branch=z9hG4bK423ec20d0a4
Call-ID: 3411a7fdb824639cd797366b423ec20d@100.1.1.188
From: <sip:2001@100.1.1.188:5060;user=phone;ttl=0>;tag=423ec20d
To: <sip:3001@100.1.1.185:5060;user=phone;ttl=0>;tag=e57f039c
CSeq: 1 INVITE
Contact: <sip:3001@100.1.1.185:5060>
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 08 Sep 2007 17:39:37 GMT
Content-Length: 167
Content-Type: application/sdp

v=0
o=HUAWEI 1073741846 1073741846 IN IP4 100.1.1.185
s=Sip Call
c=IN IP4 100.1.1.185
t=0 0
m=audio 16392 RTP/AVP 18
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no

// The server received a 183 response from the network side.

*Sep 8 17:40:32:622 2007 Sysname SSM/7/VOICE:
  SSM_PACKET [STACK]: Stack-->NetWork

SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 192.168.1.101:5060;branch=z9hG4bK275cffa6341
Call-ID: b824639c89ea2db3fc82faf275cffa660192.168.1.101
From: <sip:2001@192.168.1.101:5060;tag=275cffa6>
To: <sip:3001@100.1.1.188:5060;tag=aa465f5e>
CSeq: 1 INVITE
Contact: <sip:Sip-Server@100.1.1.188:5060;user=phone;ttl=0>
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 08 Sep 2007 17:39:37 GMT
Content-Length: 167

375
Content-Type: application/sdp

v=0
o=HUAWEI 1073741846 1073741846 IN IP4 100.1.1.185
s=Sip Call
c=IN IP4 100.1.1.185
t=0 0
m=audio 16392 RTP/AVP 18
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no

// The server sent the 183 response to the network side.

*Sep 8 17:40:42:465 2007 Sysname SSM/7/VOICE:
    SSM_PACKET [STACK]: NetWork-->Stack

SIP/2.0 200 OK
Via: SIP/2.0/UDP 100.1.1.188:5060;branch=z9hG4bK423ec20d0a4
Call-ID: 3411a7fdb824639cd797366b423ec20d@100.1.1.188
From: <sip:2001@100.1.1.188:5060;user=phone;ttl=0>;tag=423ec20d
To: <sip:3001@100.1.1.185:5060;user=phone;ttl=0>;tag=e57f039c
CSeq: 1 INVITE
Contact: <sip:3001@100.1.1.185:5060>
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 08 Sep 2007 17:39:43 GMT
Content-Length: 167
Content-Type: application/sdp

v=0
o=HUAWEI 1073741848 1073741848 IN IP4 100.1.1.185
s=Sip Call
c=IN IP4 100.1.1.185
t=0 0
m=audio 16392 RTP/AVP 18
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no

// The server received a 200 response from the network side.

*Sep 8 17:40:42:466 2007 Sysname SSM/7/VOICE:
    SSM_PACKET [STACK]: Stack-->NetWork

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.101:5060;branch=z9hG4bK275cffa6341
Call-ID: b824639c89ea2bdb3fc82f275cffa66@192.168.1.101
From: <sip:2001@192.168.1.101:5060;tag=275cffa6
To: <sip:3001@100.1.1.185:5060;tag=aa465f5e
CSeq: 1 INVITE
Contact: <sip:Sip-Server@100.1.1.188:5060;user=phone;ttl=0>
Allow: ACK,BYE,CANCEL,INFO,INVITE,NOTIFY,PRACK,REFER,REGISTER,UPDATE
Date: Sat, 08 Sep 2007 17:39:43 GMT
Content-Length: 167
Content-Type: application/sdp

v=0
o=HUAWEI 1073741848 1073741848 IN IP4 100.1.1.185
s=Sip Call
c=IN IP4 100.1.1.185
t=0 0
m=audio 16392 RTP/AVP 18
a=rtpmap:18 G729/8000
a=fmt:18 annexb=no

// The server sent the 200 response to the network side.

*Sep 8 17:40:42:484 2007 Sysname SSM/7/VOICE:
  SSM_PACKET [STACK]: NetWork--->Stack

ACK sip:Sip-Server@100.1.1.188:5060;user=phone;ttl=0 SIP/2.0
Via: SIP/2.0/UDP 100.1.1.188:5060;branch=z9hG4bK0c065f5e447
Call-ID: b824639c89ea2bdb3fc82f275cffa66@192.168.1.101
From: <sip:2001@192.168.1.101:5060>;tag=275cffa6
To: <sip:3001@100.1.1.188:5060>;tag=aa465f5e
CSeq: 1 ACK
Date: Sat, 08 Sep 2007 17:40:42 GMT
Max-Forwards: 70
Content-Length: 0

// The server received an ACK from the network side.

*Sep 8 17:40:42:484 2007 Sysname SSM/7/VOICE:
  SSM_PACKET [STACK]: Stack--->NetWork

ACK sip:3001@100.1.1.188:5060;user=phone;ttl=0 SIP/2.0
Via: SIP/2.0/UDP 100.1.1.188:5060;branch=z9hG4bK447ec20de08
Call-ID: 3411a7fadb824639c3797366b423ec20d@100.1.1.188
From: <sip:2001@100.1.1.188:5060;user=phone;ttl=0>;tag=423ec20d
To: <sip:3001@100.1.1.188:5060;user=phone;ttl=0>;tag=e57f039c
CSeq: 1 ACK
Contact: <sip:Sip-Server@100.1.1.188:5060;user=phone;ttl=0>
Date: Sat, 08 Sep 2007 17:40:42 GMT
Max-Forwards: 70
Content-Length: 0

// The server sent an ACK to the network side.

debugging voice ssm sub

Use debugging voice ssm sub to enable debugging for SIP server subscription service.
Use undo debugging voice ssm sub to disable debugging for SIP server subscription service.
Syntax

    debugging voice ssm sub { all | error | event | fsm | info | prim | timer }
    undo debugging voice ssm sub { all | error | event | fsm | info | prim | timer }

Default

Debugging for SIP server subscription service is disabled.

Views

User view

Default command level

    2: System level

Parameters

    sub: Enables debugging for SIP server subscription service.
    all: Specifies all types of debugging.
    error: Specifies error debugging.
    event: Specifies event debugging.
    fsm: Specifies finite state machine debugging.
    info: Specifies information debugging.
    prim: Specifies primitive debugging.
    timer: Specifies timer debugging.

Examples

    # Enable all types of debugging for SIP server subscription service. When User A with number 456
    # initiates a subscription to User B with number 2001 through the SIP server, output similar to the following
    # example is generated:
    <Sysname> debugging voice ssm sub all
    Enable SSM SUB ALL debugging functions
    <Sysname>
    *Sep  7 17:10:08:764 2007 Sysname SSM/7/VOICE:
        SSM_INFO [SUB_SSA]:
        General: Succeed in getting a free element from the list.
        Additional: Subscribe control block ID is 0.
    // The SSA module got a free element from the list.
    *Sep  7 17:10:08:764 2007 Sysname SSM/7/VOICE:
        SSM_INFO [SUB_SSA]:
        General: Received message from the stack.
        Additional: The message type is ReferInd.
        Subscribe control block ID is 0
        Caller number is 456
        Called number is Sip-Server
    // The SSA module received a ReferInd message from the protocol stack.
    *Sep  7 17:10:08:764 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Send UCM Subscribe request message.

// The SSA module sent a subscription request to the UCM module.

*Sep 7 17:10:08:765 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Receive UCM Subscribe request message.

// The SLC module received the subscription request from the UCM module.

*Sep 7 17:10:08:765 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SLC]:
General: Succeed in getting a free element from the list.
Additional: Subscribe control block ID is 0.

// The SLC module got a free element from the list.

*Sep 7 17:10:08:765 2007 Sysname SSM/7/VOICE:
SSM_EVENT [SUB_SLC]:
SUBSCRIBE: Received event SUBSCRIBEREQ while at state IDLE.(ObjectID = 0)

// The SLC module received a SUBSCRIBEREQ event in the IDLE state.

*Sep 7 17:10:08:766 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SLC]:
General: Succeed in associate call.
Additional: Subscribe control block ID is 0
Call control block ID is 6
Caller number is 456
Called number is Sip-Server

// The SLC module found the relevant session.

*Sep 7 17:10:08:766 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Send UCM Subscribe request message.

// The SLC module sent a subscription request to the UCM module.

*Sep 7 17:10:08:766 2007 Sysname SSM/7/VOICE:
SSM_FSM [SUB_SLC]:
SUBSCRIBE: State changed from IDLE to SUB.(ObjectID = 0)

// The SLC module changed from IDLE state to SUB state.

*Sep 7 17:10:08:963 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Receive UCM Subscribe request message.

// The SSA module received the subscription request from the UCM module.

*Sep 7 17:10:09:273 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SSA]:
General: Succeed in getting a free element from the list.
Additional: Subscribe control block ID is 1.

// The SSA module got a free element from the list.

*Sep  7 17:10:09:373 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SSA]:
General: Succeed in sending the network message.
Additional: Method type is REFER.

// The SSA module sent a REFER message to the network side.

*Sep  7 17:10:09:473 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SSA]:
General: Received message from the stack.
Additional: The message type is ReferCfm.

*Sep  7 17:10:09:673 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Send UCM Subscribe response message.

*Sep  7 17:10:09:823 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Receive UCM Subscribe response message.

*Sep  7 17:10:10:73 2007 Sysname SSM/7/VOICE:
SSM_EVENT [SUB_SLC]:
SUBSCRIBE: Received event SUBSCRIBECFM while at state SUB.(ObjectID = 0)

*Sep  7 17:10:10:173 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Send UCM Subscribe response message.

*Sep  7 17:10:10:323 2007 Sysname SSM/7/VOICE:
SSM_TIMER [SUB_SLC]:
Start Timer: Type = 7, Group = 7, Index = 0, Length = 120000

// The SLC module started a timer, with the value set to 120000 ms.

*Sep  7 17:10:10:473 2007 Sysname SSM/7/VOICE:
SSM_FSM [SUB_SLC]:
SUBSCRIBE: State changed from SUB to ACTIVE.(ObjectID = 0)

// The SLC module changed from SUB state to ACTIVE state.

*Sep  7 17:10:10:623 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Receive UCM Subscribe response message.

*Sep  7 17:10:10:833 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SSA]:
General: Succeed in sending the network message.
Additional: Method type is REFER, status code is 202.

*Sep 7 17:10:10:983 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SSA]:
General: Received message from the stack.
Additional: The message type is NotifyInd.

*Sep 7 17:10:11:83 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Send UCM Subscribe request message.

*Sep 7 17:10:11:233 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Receive UCM Subscribe request message.

*Sep 7 17:10:11:383 2007 Sysname SSM/7/VOICE:
SSM_EVENT [SUB_SLC]:
SUBSCRIBE: Received event SUBREQUESTREQ while at state ACTIVE.(ObjectID = 0)

*Sep 7 17:10:11:483 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Send UCM Subscribe request message.

*Sep 7 17:10:11:633 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Receive UCM Subscribe request message.

*Sep 7 17:10:11:783 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SSA]:
General: Succeed in sending the network message.
Additional: Method type is NOTIFY.

*Sep 7 17:10:11:883 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SSA]:
General: Received message from the stack.
Additional: The message type is NotifyCfm.

// The SSA module received a NotifyCfm message from the protocol stack.

*Sep 7 17:10:12:83 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Send UCM Subscribe response message.

*Sep 7 17:10:12:243 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Receive UCM Subscribe response message.
SSM_EVENT [SUB_SLC]:
SUBSCRIBE: Received event SUBREQUESTRSP while at state ACTIVE.(ObjectID = 0)

SSM_PRIM [SUB_SLC]:
Send UCM Subscribe response message.

SSM_PRIM [SUB_SSA]:
Receive UCM Subscribe response message.

// The SSA module received a response to the subscription request from the UCM module.

SSM_INFO [SUB_SSA]:
General: Succeed in sending the network message.
Additional: Method type is NOTIFY, status code is 200.

SSM_INFO [SUB_SSA]:
General: Received message from the stack.
Additional: The message type is NotifyInd.

SSM_PRIM [SUB_SSA]:
Send UCM Subscribe request message.

SSM_PRIM [SUB_SLC]:
Receive UCM Subscribe request message.

SSM_EVENT [SUB_SLC]:
SUBSCRIBE: Received event SUBREQUESTRSP while at state ACTIVE.(ObjectID = 0)

SSM_PRIM [SUB_SLC]:
Send UCM Subscribe request message.

SSM_FSM [SUB_SLC]:
SUBSCRIBE: State changed from ACTIVE to TERMINAL.(ObjectID = 0)

SSM_PRIM [SUB_SSA]:
Receive UCM Subscribe request message.
SSM_INFO [SUB_SSA]:
General: Succeed in sending the network message.
Additional: Method type is NOTIFY.

*Sep  7 17:10:17:739 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SSA]:
General: Received message from the stack.
Additional: The message type is NotifyCfm.

*Sep  7 17:10:17:935 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Send UCM Subscribe response message.

*Sep  7 17:10:18:35 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Receive UCM Subscribe response message.

*Sep  7 17:10:18:185 2007 Sysname SSM/7/VOICE:
SSM_EVENT [SUB_SLC]:
SUBSCRIBE: Received event SUBREQUESTRSP while at state TERMINAL.(ObjectID = 0)

*Sep  7 17:10:18:335 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Send UCM Subscribe response message.

*Sep  7 17:10:18:485 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Send UCM Subscribe release message.

*Sep  7 17:10:18:585 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SLC]:
Send UCM Subscribe release message.

*Sep  7 17:10:18:735 2007 Sysname SSM/7/VOICE:
SSM_TIMER [SUB_SLC]:
Stop Timer: Type = 7, Group = 7, Index = 0, Length = 120000

// The SLC module stopped a 120000-ms timer.

*Sep  7 17:10:18:885 2007 Sysname SSM/7/VOICE:
SSM_INFO [SUB_SLC]:
General: Succeed in freeing element to the list.
Additional: Subscribe control block ID is 0.

// The SSA module released an element to the list.

*Sep  7 17:10:19:85 2007 Sysname SSM/7/VOICE:
SSM_PRIM [SUB_SSA]:
Receive UCM Subscribe response message.
// The SSA module received an unsubscription message from the UCM module.

// The SSA module sent a SUBRELEASE message to the network side.

debugging voice ssm ucm

Use debugging voice ssm ucm to enable debugging for SIP server call management.

Use undo debugging voice ssm ucm to disable debugging for SIP server call management.

Syntax

debugging voice ssm ucm { all | error | info }
undo debugging voice ssm ucm { all | error | info }

Default

Debugging for SIP server call management is disabled.
Views

User view

Default command level

2: System level

Parameters

ucm: Enables debugging for SIP server call management.
all: Specifies all types of debugging.
error: Specifies error debugging.
info: Specifies information debugging.

Examples

# Enable all types of debugging for SIP server call management.
<Sysname> debugging voice ssm ucm all
Enable SSM UCM ALL debugging functions

- When the server enable command is used to enable the SIP server, output similar to the following example is generated:
  * Sep 8 16:59:38:367 2007 Sysname SSM/7/VOICE:
    SSM_INFO [UCM]:
    General: Succeed in starting the server.
    // The UCM module started the server.

- When the undo server enable command is used to disable the SIP server, output similar to the following example is generated:
  * Sep 8 17:00:13:728 2007 Sysname SSM/7/VOICE:
    SSM_INFO [UCM]:
    General: Succeed in stopping the server.
    // The UCM module stopped the server.

debugging voice srtp

Use debugging voice srtp to enable debugging for SRTP messages.
Use undo debugging voice srtp to disable debugging for SRTP messages.

Syntax

debugging voice srtp { all | error | event }
undo debugging voice srtp { all | error | event }

Default

Debugging for SRTP messages is disabled.

Views

User view

Default command level

2: System level
Parameters

srtp: Enables debugging for SRTP messages.
all: Specifies debugging for all SRTP message types.
error: Specifies debugging for error messages.
event: Specifies debugging for event messages.

Examples

# Enable debugging for SRTP event messages. When a call is established by using SRTP, output similar to the following example is generated:

*Apr 10 09:36:53:924 2010 H3C_Down SRTP/7/VOICE:
SRTP_EVENT: Derive the session key with AES_ICM algorithm.

// SRTP used the AES-ICM algorithm to generate a session key.

*Apr 10 09:36:53:924 2010 H3C_Down SRTP/7/VOICE:
SRTP_EVENT: AES_ICM: The context has been initialized.

// SRTP initialized the context for the AES-ICM algorithm.

*Apr 10 09:36:53:926 2010 H3C_Down SRTP/7/VOICE:
SRTP_EVENT: The cipher type is AES_ICM, generate RTP salt key.

// The encryption algorithm was AES_ICM, and an RTP salt key was generated.

*Apr 10 09:36:53:926 2010 H3C_Down SRTP/7/VOICE:
SRTP_EVENT: AES_ICM: The context has been initialized.

// SRTP initialized the context for the AES-ICM algorithm.
Smart Link debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

The Smart Link module name is identified as "SMLK" in debugging messages.

debugging smart-link

Use `debugging smart-link` to enable debugging for the specified or all smart link groups.
Use `undo debugging smart-link` to disable debugging for the specified or all smart link groups.

Syntax

```
debugging smart-link [ group group-id ] { all | error | event | fsm | packet }
undo debugging smart-link [ group group-id ] { all | error | event | fsm | packet }
```

Default

Smart link group debugging is disabled.

Views

User view

Default command level

1. Monitor level

Parameters

- **group group-id**: Specifies a smart link group by its number.
- **all**: Specifies all Smart Link debugging.
- **error**: Specifies Smart Link error debugging.
- **event**: Specifies Smart Link event debugging.
- **fsm**: Specifies state machine debugging.
- **packet**: Specifies packet debugging.

Usage guidelines

Table 1 describes output fields and messages for the `debugging smart-link error` command.

Table 195 Output from the debugging smart-link error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Failed to allocate memory for String | SMLK failed to allocate memory for the following:  
  - batch bak—Bulk backup.  
  - realtime bak—Realtime backup.  
  - create group—Smart link group creation.  
  - smlk port info—Smart link port information. |
| Failed to create mbuffer | Creating MBUF failed.                                                        |
Table 2 describes output fields and messages for the **debugging smart-link packet** command.

**Table 196 Output from the debugging smart-link packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Name : String</td>
<td>A Flush packet was sent or received on the port identified by Port Name.</td>
</tr>
<tr>
<td>VLAN Bit Map</td>
<td>VLAN bit map, presenting the list of VLANs carried on the other port in the smart link group.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable smart link group error debugging on a device with the Flush message reception capability. When you configure the receive-control VLAN on Ethernet 1/1 to be different from the control VLAN of the communicating smart link group, output similar to the following example is generated:

```
<Sysname> debugging smart-link error
*May 15 15:34:06:931 2007 Sysname SMLK/7/RCVPKterr:Slot=2;
   Received packet on port Ethernet1/1 error for control VLAN is different
```

// An error occurred because SMLK received a Flush message on port Ethernet 1/1. The message includes a control VLAN different from the control VLAN for the receive-control VLAN configured on the port.

# Enable event debugging for smart link group 1. The output in this example was created when the following conditions exist:

- Create smart link group 1.
- Assign the master and backup ports to smart link group 1.

When a link switchover occurred in the group, output similar to the following example is generated:

```
<Sysname> debugging smart-link group 1 event
*May 15 15:35:12:427 2007 Sysname SMLK/7/SMLKSEND:
   Smlk group 1 sent packet started.
*May 15 15:35:12:437 2007 Sysname SMLK/7/SMLKSEND:
   Smlk group 1 sent packet finished.
```

// Smart link group 1 started to send packets upon a link switchover.

// Smart link group 1 finished sending packets.

# Enable FSM debugging for smart link group 1. The output in this example was created when the following conditions exist:

- Create smart link group 1.
- Assign port Ethernet 1/1 to the group as the master port.

```
<Sysname> debugging smart-link group 1 fsm
*May 15 15:58:13:572 2007 Sysname SMLK/7/SMLKFsm:
   Smart link group 1 : PIM occur event: Enter PIM state machine.
*May 15 15:58:13:581 2007 Sysname SMLK/7/SMLKFsm:
   Smart link group 1 Port Ethernet1/1: PIM occur event: Enter EVENT_MASTER case.
*May 15 15:58:13:592 2007 Sysname SMLK/7/SMLKFsm:
```

388
Smart link group 1 : PSS occur event: Enter PSS state machine.
*May 15 15:58:13:602 2007 Sysname SMLK/7/SMLKFSM:
Smart link group 1 : PSS occur event: Member number is 1.
*May 15 15:58:13:611 2007 Sysname SMLK/7/SMLKFSM:
Smart link group 1 Port Ethernet1/1: PSS occur event: Link status is up.
*May 15 15:58:13:622 2007 Sysname SMLK/7/SMLKFSM:
Smart link group 1 Port Ethernet1/1: PST occur event: Enter PST state machine.
*May 15 15:58:13:632 2007 Sysname SMLK/7/SMLKFSM:
Smart link group 1 Port Ethernet1/1: PST occur event: State changes to active.

# Enable packet debugging for smart link group 1. The output in this example was created when the
following conditions exist:
•

Create smart link group 1.

•

Assign the master and backup ports to smart link group 1.

<Sysname> debugging smart-link group 1 packet
*May 15 16:04:30:546 2007 Sysname SMLK/7/SMLKPKT:
Smart link group 1:
Port Ethernet1/1: Send Flush Packet

// Port Ethernet 1/1 sent a Flush message. This indicates that a link switchover occurred in the group.
Device ID: 000f-e200-8500

// The device ID is 000F-E200-8500.
Control VLAN: 1

// The control VLAN is VLAN 1.
VLAN Bit Map:
02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*May 15 16:04:30:625 2007 Sysname SMLK/7/SMLKPKT:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

389


SNMP debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging snmp agent packet

Use `debugging snmp agent packet` to enable SNMP packet debugging for the SNMP agent.

Use `undo debugging snmp agent packet` to disable SNMP packet debugging for the SNMP agent.

Syntax

```
debugging snmp agent packet { header | receive | send }
undo debugging snmp agent packet { header | receive | send }
```

Default

SNMP packet debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- **header**: Enables SNMP packet header debugging to output to the information center the following information from an SNMP request header: version, community name, and username.

- **receive**: Enables debugging for the SNMP packet received to output to the information center the following information from an SNMP request received by the SNMP agent: the packet type, request-id, error-status, error-index, and the variable binding list.

- **send**: Enables debugging for the SNMP packet sent to output to the information center the following information from an SNMP response sent by the SNMP agent: packet type, request-id, error-status, error-index, and the variable binding list.

Usage guidelines

Table 1 describes output fields and messages for the `debugging snmp agent packet header` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>incoming SNMP-version packet</td>
<td>An SNMP packet was received. SNMP-version represents the SNMP version, SNMPv1, SNMPv2c, or SNMPv3.</td>
</tr>
<tr>
<td>community name: community-name</td>
<td>SNMPv1/v2c community name.</td>
</tr>
<tr>
<td>security model: v3</td>
<td>SNMPv3 security model.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>security level:</td>
<td>SNMPv3 security level:</td>
</tr>
<tr>
<td>security-level</td>
<td>• noAuthNoPriv—No authentication, no privacy.</td>
</tr>
<tr>
<td></td>
<td>• authNoPriv—Authentication without privacy.</td>
</tr>
<tr>
<td></td>
<td>• authPriv—Authentication with privacy.</td>
</tr>
<tr>
<td>user name:</td>
<td>SNMPv3 username.</td>
</tr>
<tr>
<td>user-name</td>
<td></td>
</tr>
<tr>
<td>snmpEngineID:</td>
<td>SNMP engine ID.</td>
</tr>
<tr>
<td>engineID</td>
<td></td>
</tr>
<tr>
<td>snmpEngineBoots:</td>
<td>Number of SNMP engine reboots.</td>
</tr>
<tr>
<td>n</td>
<td></td>
</tr>
<tr>
<td>snmpEngineTime:</td>
<td>SNMP engine running time in seconds.</td>
</tr>
<tr>
<td>n</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the **debugging snmp agent packet receive** command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACKET</td>
<td>Information contained in the SNMP packet.</td>
</tr>
<tr>
<td>PACKET_SRC</td>
<td>Source address information for the SNMP packet.</td>
</tr>
<tr>
<td>packet received from address via UDP</td>
<td>The SNMP packet was received from address through UDP. address represents the source IP address of the SNMP packet.</td>
</tr>
<tr>
<td>request-id:</td>
<td>ID of the SNMP request, used to match the SNMP response.</td>
</tr>
<tr>
<td>request-id</td>
<td></td>
</tr>
<tr>
<td>error-status:</td>
<td>Error status of the SNMP request.</td>
</tr>
<tr>
<td>error-status</td>
<td></td>
</tr>
<tr>
<td>error-index:</td>
<td>Error index of the SNMP request.</td>
</tr>
<tr>
<td>error-index</td>
<td></td>
</tr>
<tr>
<td>VBLIST</td>
<td>Variable binding list.</td>
</tr>
<tr>
<td>get request</td>
<td>SNMP get request.</td>
</tr>
<tr>
<td>set request</td>
<td>SNMP set request.</td>
</tr>
<tr>
<td>get-next request</td>
<td>SNMP get-next request.</td>
</tr>
<tr>
<td>get-bulk request</td>
<td>SNMP get-bulk request.</td>
</tr>
<tr>
<td>non-repeaters:</td>
<td>The non-repeaters field of a get-bulk request.</td>
</tr>
<tr>
<td>non-repeaters</td>
<td></td>
</tr>
<tr>
<td>max-repetitions:</td>
<td>The max-repetitions field of a get-bulk request.</td>
</tr>
<tr>
<td>max-repetitions</td>
<td></td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the **debugging snmp agent packet send** command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACKET</td>
<td>Information contained in the SNMP packet.</td>
</tr>
<tr>
<td>PACKET_DES</td>
<td>Destination address information for the SNMP packet.</td>
</tr>
<tr>
<td>packet sent to address via UDP</td>
<td>The SNMP packet was sent to address through UDP. address represents the destination IP address of the SNMP packet.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>request-id: request-id</td>
<td>ID of the SNMP response, used to match the SNMP request.</td>
</tr>
<tr>
<td>error-status: error-status</td>
<td>Error status of the SNMP response.</td>
</tr>
<tr>
<td>error-index: error-index</td>
<td>Error index of the SNMP response.</td>
</tr>
<tr>
<td>VLIST</td>
<td>Variable binding list.</td>
</tr>
<tr>
<td>response</td>
<td>SNMP response.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable information center debugging and SNMP packet header debugging on the device. Output similar to the following example is generated when network management software is used to access the device under these conditions:

- SNMPv1 is enabled.
- A community name is configured.

```
<Sysname> terminal debugging
% Current terminal debugging is on
<Sysname> terminal monitor
% Current terminal monitor is on
<Sysname> debugging snmp agent packet header
*Jul 27 08:37:26:313 2007 Sysname SNMP/7/HEADER:
  incoming SNMPv1 packet
  community name: public
```

// SNMP received an SNMPv1 request with the community name **public**.

# Enable information center debugging and SNMP packet header debugging on the device. Output similar to the following example is generated when network management software is used to access the device under these conditions:

- SNMPv2c is enabled.
- A community name is configured.

```
<Sysname> terminal debugging
% Current terminal debugging is on
<Sysname> terminal monitor
% Current terminal monitor is on
<Sysname> debugging snmp agent packet header
*Jul 27 08:37:26:313 2007 Sysname SNMP/7/HEADER:
  incoming SNMPv2c packet
  community name: private
```

// SNMP received an SNMPv2c request with the community name **private**.

# Enable information center debugging and SNMP packet header debugging on the device. Output similar to the following example is generated when network management software is used to access the device under these conditions:

- SNMPv3 is enabled.
- A community name is configured.

```
<Sysname> terminal debugging
% Current terminal debugging is on
```
<Sysname> terminal monitor
% Current terminal monitor is on
<Sysname> debugging snmp agent packet header
*Jul 27 08:51:00:563 2007 Sysname SNMP/7/HEADER:
  incoming SNMPv3 packet
  security model: v3
  security level: authNoPriv
  user name: v3user1
  snmpEngineID: 000063A27F00000100001707
  snmpEngineBoots: 1
  snmpEngineTime: 54591

// SNMP received an SNMPv3 request.

# Enable information center debugging and received SNMP packet debugging on the device. Output similar to the following example is generated when network management software is used to perform a Get operation on object sysUpTime.0 under these conditions:
- SNMPv2c is enabled.
- A community name is configured.

<Sysname> terminal debugging
% Current terminal debugging is on
<Sysname> terminal monitor
% Current terminal monitor is on
<Sysname> debugging snmp agent packet receive
*Jul 27 08:58:52:594 2007 Sysname SNMP/7/PACKET_SRC:
  packet received from 10.165.81.75 via UDP
*Jul 27 08:58:52:594 2007 Sysname SNMP/7/PACKET:
  get request
    request-id: 13
    error-status: 0
    error-index: 0
*Jul 27 08:58:52:594 2007 Sysname SNMP/7/VBLIST:
  sysUpTime.0:

// SNMP received an SNMP request from 10.165.81.75 through UDP.

# Enable information center debugging and sent SNMP packet debugging on the device. Output similar to the following example is generated when network management software is used to perform a Get operation on object sysUpTime.0 under these conditions:
- SNMPv2c is enabled.
- A community name is configured.

<Sysname> terminal debugging
% Current terminal debugging is on
<Sysname> terminal monitor
% Current terminal monitor is on
<Sysname> debugging snmp agent packet send
*Jul 27 09:08:21:563 2007 Sysname SNMP/7/PACKET:
  response
    request-id: 16
    error-status: 0
debugging snmp agent process

Use **debugging snmp agent process** to enable debugging for SNMP packet processing.

Use **undo debugging snmp agent process** to disable debugging for SNMP packet processing.

**Syntax**

```
debugging snmp agent process { all | decode | stack | txrx } { error | info | warning }
undo debugging snmp agent process { all | decode | stack | txrx } { error | info | warning }
```

**Default**

SNMP packet processing debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **all**: Enables debugging for all SNMP packet processing, including **decode**, **stack**, and **txrx**.
- **decode**: Enables debugging for decoding an SNMP request.
- **stack**: Enables debugging for handling the SNMP request PDU.
- **txrx**: Enables debugging for transmitting or receiving an SNMP message.
- **error**: Enables error-level debugging to output error debugging information, which indicates the error information generated during the running of the SNMP stack or the system.
- **info**: Enables info-level debugging to output informational debugging information, which indicates the prompt information generated during the running of the SNMP stack or the system.
- **warning**: Enables warning-level debugging to output warning debugging information, which indicates the important information generated during the running of the SNMP stack or the system.

**Usage guidelines**

If you do not specify the level of the debugging information, debugging information for all levels will be output or disabled.

**Table 4** describes output fields and messages for the **debugging snmp agent process decode** command.

**Table 200 Output from the debugging snmp agent process decode command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECODE_INFO</td>
<td>Info-level debugging information generated during the SNMP request decoding process.</td>
</tr>
</tbody>
</table>
Table 5 describes output fields and messages for the **debugging snmp agent process txrx** command.

Table 201 Output from the debugging snmp agent process txrx command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECODE_WARNING</td>
<td>Warning-level debugging information generated during the SNMP request decoding process.</td>
</tr>
<tr>
<td>DECODE_ERROR</td>
<td>Error-level debugging information generated during the SNMP request decoding process.</td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the **debugging snmp agent process stack** command.

Table 202 Output from the debugging snmp agent process stack command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXRX_INFO</td>
<td>Info-level debugging information generated when SNMP messages were being sent and received.</td>
</tr>
<tr>
<td>TXRX_WARNING</td>
<td>Warning-level debugging information generated when SNMP messages were being sent and received.</td>
</tr>
<tr>
<td>TXRX_ERROR</td>
<td>Error-level debugging information generated when SNMP messages were being sent and received.</td>
</tr>
<tr>
<td>failed to set IPv6 socket option while creating IPv6 socket (errCode: error-code)</td>
<td>This IPv6 socket option determines whether the IP address and port number used by this socket can be bound with another socket.</td>
</tr>
<tr>
<td>failed to set IPv4 socket option while receiving PDU through IPv4 socket</td>
<td>This IPv4 socket option determines whether the destination IP address can be parsed out from the received packet.</td>
</tr>
</tbody>
</table>

Examples

# Enable information center debugging and info-level SNMP PDU debugging on the device. Output similar to the following example is generated when network management software is used to perform a Get operation on object `sysUpTime` under these conditions:
- SNMPv2c is enabled.
- A community name is configured.

SYSNAME> terminal debugging
% Current terminal debugging is on
SYSNAME> terminal monitor
% Current terminal monitor is on
SYSNAME> debugging snmp agent process stack info
SYSNAME>

* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  read request
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  PDU type is 160 and PDU version is 1.
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  make request
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  process request
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  get instance sysUpTime.0
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  create CFG message for get-request and parse CFG message for response
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  create CFG message
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  create CFG message for leaf node
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  get instance sysUpTime.0 in batch way
* Jul 27 09:42:13:578 2007 Sysname SNMP/7/STACK_INFO:
  parse CFG message and fill variable-bindings
* Jul 27 09:42:13:594 2007 Sysname SNMP/7/STACK_INFO:
  get value from CFG message (CMO type: 3)
* Jul 27 09:42:13:594 2007 Sysname SNMP/7/STACK_INFO:
  convert data type 43 from CFG message
* Jul 27 09:42:13:594 2007 Sysname SNMP/7/STACK_INFO:
  fill variable-bindings
* Jul 27 09:42:13:594 2007 Sysname SNMP/7/STACK_INFO:
  send response to NMS (errStatus: 0, errIndex: 0)

// Info-level debugging information was generated during the process of handling the SNMP request
PDU and generating the response.

debugging snmp trap packet

Use debugging snmp trap packet to enable debugging for SNMP trap messages.
Use undo debugging snmp trap packet to disable debugging for SNMP trap messages.

Syntax

debugging snmp trap packet
undo debugging snmp trap packet
Default

Debugging for SNMP trap messages is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

Table 7 describes output fields and messages for the **debugging snmp trap packet** command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAP_PACKET</td>
<td>Trap message debugging information.</td>
</tr>
<tr>
<td>trap-name</td>
<td>The system sent the <em>trap-name</em> trap message to <em>address</em>.</td>
</tr>
<tr>
<td>version</td>
<td>• <em>trap-name</em>—Name of the trap message.</td>
</tr>
<tr>
<td></td>
<td>• <em>version</em>—Trap version, which is v1 or v2c.</td>
</tr>
<tr>
<td>address</td>
<td>• <em>address</em>—Destination IP address.</td>
</tr>
<tr>
<td>request-id</td>
<td>The <em>request-id</em> field of the trap PDU. The value of the field is always 0.</td>
</tr>
<tr>
<td>error-status</td>
<td>The <em>error-status</em> field of the trap PDU. The value of the field is always 0.</td>
</tr>
<tr>
<td>error-index</td>
<td>The <em>error-index</em> field of the trap PDU. The value of the field is always 0.</td>
</tr>
<tr>
<td>UDP port</td>
<td>UDP port number of the destination host for receiving the trap messages.</td>
</tr>
<tr>
<td>send OK</td>
<td>Traps were sent successfully.</td>
</tr>
<tr>
<td>VBLIST</td>
<td>Variable binding list in the trap message.</td>
</tr>
</tbody>
</table>

Examples

# Enable information center debugging and SNMP trap message debugging on the device enabled with the SNMP trap sending function. When the undo snmp-agent command and the snmp-agent command are executed in system view, output similar to the following example is generated:

```
<Sysname> terminal debugging
% Current terminal debugging is on
<Sysname> terminal monitor
% Current terminal monitor is on
<Sysname> debugging snmp trap packet
<Sysname> undo snmp-agent
[Sysname] snmp-agent
[Sysname]
*Jul 27 10:10:35:297 2007 Sysname SNMP/7/TRAP_PACKET:
  warmStart trap<v2> send to: 10.165.81.75
  request-id: 0
  error-status: 0
  error-index: 0
  UDP port: 162
```
// The device sent the version 2 trap message to the destination host 10.165.81.75.
The trap message was sent successfully, and the values of the binding variables sysUpTime.0 and snmpTrapOID.0 are 5936387 and 1.3.6.1.6.3.1.1.5.2, respectively.

debugging snmp trap process

Use **debugging snmp trap process** to enable debugging for the SNMP trap task.

Use **undo debugging snmp trap process** to disable debugging for the SNMP trap task.

**Syntax**

```
debugging snmp trap process [ error | info | warning ]
undo debugging snmp trap process [ error | info | warning ]
```

**Default**

Debugging for the SNMP trap task is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **error**: Specifies error debugging for the SNMP trap task.
- **info**: Specifies info debugging for the SNMP trap task.
- **warning**: Specifies warning debugging for the SNMP trap task.

**Usage guidelines**

If you do not specify the level of debugging information, debugging information for all levels will be output or disabled.

**Table 8** describes output fields and messages for the **debugging snmp trap process** command.

**Table 204 Output from the debugging snmp trap process command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAP_INFO</td>
<td>Info-level debugging information generated during the trap task implementation process.</td>
</tr>
<tr>
<td>TRAP_WARNING</td>
<td>Warning-level debugging information generated during the trap task implementation process.</td>
</tr>
<tr>
<td>TRAP_ERROR</td>
<td>Error-level debugging information generated during the trap task implementation process.</td>
</tr>
<tr>
<td>wrong number of parameters in trap-name's trap message</td>
<td>Number of parameters in the trap message is incorrect.</td>
</tr>
</tbody>
</table>
### Field Description

| filter address address for sending trap-name | Trap messages whose destination address is address are filtered when trap messages are being sent.
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• trap-name—Name of the trap node.</td>
<td></td>
</tr>
<tr>
<td>• address—Destination IP address of trap messages.</td>
<td></td>
</tr>
</tbody>
</table>

### Examples

# Enable information center debugging and SNMP trap module debugging on the device enabled with the SNMP trap sending function. When the undo snmp-agent command and the snmp-agent command are executed in system view, output similar to the following example is generated:

```
<Sysname> terminal debugging
% Current terminal debugging is on
<Sysname> terminal monitor
% Current terminal monitor is on
<Sysname> debugging snmp trap process info
[Sysname] undo snmp-agent
[Sysname] snmp-agent
[Sysname]
*Jul 27 10:21:22:984 2007 Sysname SNMP/7/TRAP_INFO: search entries in snmpTargetAddrTable to match snmpNotifyTag TrapHost
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO: search entries in snmpTargetParamsTable to match snmpTargetAddrParams traphost.u2.192.168.123.123
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO: check trap warmStart's VACM
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO: get source IPv4 address for sending trap
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO: 400
```
search entries in snmpTargetParamsTable to match snmpTargetAddrParams
traphost.uu.10.165.81.75
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO:
  check trap warmStart's VACM
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO:
  get source IPv4 address for sending trap
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO:
  create warmStart packet
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO:
  get enterprise OID
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO:
  encode SNMPv2c trap
*Jul 27 10:21:23:00 2007 Sysname SNMP/7/TRAP_INFO:
  send trap through IPv4 socket
*Jul 27 10:21:23:16 2007 Sysname SNMP/7/TRAP_INFO:
  send warmStart successfully at 10:21:22

// The SNMP trap module handled the trap task, output the info-level debugging information, and sent
the warmStart trap message.
Spanning tree debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging stp all

Use `debugging stp all` to enable all types of STP debugging.
Use `undo debugging stp all` to disable all types of STP debugging.

Syntax

depugging stp all
undo debugging stp all

Default

All types of STP debugging are disabled.

Views

User view

Default command level

2: System level

Examples

# Enable all types of STP debugging.
<Sysname> debugging stp all

debugging stp event

Use `debugging stp event` to enable STP event debugging.
Use `undo debugging stp event` to disable STP event debugging.

Syntax

depugging stp event [ interface interface-type interface-number ]
undo debugging stp event [ interface interface-type interface-number ]

Default

STP event debugging is disabled.

Views

User view

Default command level

2: System level

Parameters

`interface interface-type interface-number`: Specifies a port by its type and number. If you do not specify a port, this command enables STP event debugging for all ports.
Usage guidelines

Table 1 describes output fields and messages for the `debugging stp event` command.

**Table 205 Output from the debugging stp event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port PortID</td>
<td>A String event occurred on port PortID. The possible values of the String argument are:</td>
</tr>
<tr>
<td></td>
<td>• ADD VLAN—The port was assigned to a VLAN.</td>
</tr>
<tr>
<td></td>
<td>• DEL VLAN—The port was removed from a VLAN.</td>
</tr>
<tr>
<td></td>
<td>• SPEED CHANGE—The port speed changed.</td>
</tr>
<tr>
<td></td>
<td>• DUPLEX CHANGE—The duplex mode of the port changed.</td>
</tr>
<tr>
<td></td>
<td>• LINK DOWN—A link-down event occurred on the port.</td>
</tr>
<tr>
<td></td>
<td>• LINK UP—A link-up event occurred on the port.</td>
</tr>
</tbody>
</table>

Examples

```
# Enable STP event debugging for port Ethernet 1/1.
<Sysname> debugging stp event interface ethernet 1/1
*Mar 18 14:28:41:887 2010 Sysname MSTP/7/PEVT: Slot=1;Port 2(Ethernet1/1) occurs LINK DOWN event
// A link-down event occurred on Ethernet 1/1.
```

debugging stp fsm

Use `debugging stp fsm` to enable STP finite state machine (FSM) debugging.

Use `undo debugging stp fsm` to disable STP FSM debugging.

**Syntax**

```
debugging stp fsm [ instance instance-id | vlan vlan-id ] [ interface interface-type interface-number ]
undo debugging stp fsm [ instance instance-id | vlan vlan-id ] [ interface interface-type interface-number ]
```

**Default**

STP FSM debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- `instance instance-id`: Specifies an MSTI. The `instance-id` argument represents the MSTI ID. The minimum value of this argument is 0, which represents the common internal spanning tree (CIST). The maximum value depends on the device model. If you do not specify an MSTI, this command enables STP FSM debugging for all MSTIs. This option is not applicable to the PVST mode.

- `vlan vlan-id`: Specifies a VLAN by its ID. The value range for the `vlan-id` argument varies with device models. If you do not specify a VLAN, this command enables STP FSM debugging for all VLANs. This option is only applicable to the PVST mode.
**interface interface-type interface-number:** Specifies a port by its type and number. If you do not specify a port, this command enables STP FSM debugging for all ports.

**Usage guidelines**

Table 2 describes output fields and messages for the `debugging stp fsm` command.

**Table 206 Output from the debugging stp fsm command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instance</strong> InstanceID’s port</td>
<td>PortID(PortName) enters String state</td>
</tr>
<tr>
<td><strong>VLAN</strong> VLANID’s port</td>
<td>PortID(PortName) enters String state</td>
</tr>
</tbody>
</table>

The state of port PortID(PortName) on MSTI InstanceID or VLAN VLANID is String. The possible values of the String argument are:

- PIM%ENABLED
- PIM%AGED
- PIM%CURRENT
- PIM%RECEIVED
- PIM%SUPERIOR_DESIGNATED
- PIM%REPEATED_DESIGNATED
- PIM%ROOT
- PIM%OTHER
- PPM%SEND_RSTP
- PPM%SENDING_RSTP
- PPM%SEND_STP
- PPM%SENDING_STP
- PRT%BLOCK_PORT
- PRT%BLOCKED_PORT
- PRT%BACKUP_PORT
- PRT%ACTIVE_PORT
- PRT%PROPOSED
- PRT%PROPOSING
- PRT%AGREES
- PRT%SYNCHED
- PRT%LEARN
- PRT%FORWARD
- PRT%REROOT
- PRT%LISTEN
- PRT%REROOTED
- PRT%ROOT
- TCM%INIT
- TCM%INACTIVE
- TCM%DETECTED
- TCM%ACTIVE
- TCM%NOTIFIED_TC
- TCM%NOTIFIED_TC
- TCM%PROPAGATING
- TCM%ACKNOLEDGED.

The percent sign (%) is preceded by the state machine name and followed by the state machine state.
### Field | Description
---|---
Instance InstanceID's port PortID(PortName) is selected as String role
VLAN VLANID's port PortID(PortName) is selected as String role

The role of port PortID(PortName) on MSTI InstanceID or VLAN VLANID is String. The possible values of the String argument are:
- **DESIGNATED**—The port is a designated port.
- **ROOT**—The port is a root port.
- **ALTERNATE**—The port is an alternate port.
- **BACKUP**—The port is a backup port.
- **MASTER**—The port is a master port.

### Examples

# In MSTP mode, enable STP FSM debugging for all ports in MSTI 2.
```
<Sysname> debugging stp fsm instance 2
```
*Mar 18 14:28:41:739 2010 Sysname MSTP/7/MEXS:Slot=1;Instance 2's port105(Ethernet1/1) enters PTX%PERIODIC state.

**// Ethernet 1/1’s PTX state machine in MSTI 2 was moved to the PERIODIC state.**
```
*Mar 18 14:28:41:739 2010 Sysname MSTP/7/MEXS:Slot=1;Instance 2's port105(Ethernet1/1) enters PTX%MSTP state.
```
**// Ethernet 1/1’s PTX state machine in MSTI 2 was moved to the MSTP state.**
```
*Mar 18 14:28:41:741 2010 Sysname MSTP/7/MEXS:Slot=1;Instance 2's port105(Ethernet1/1) enters PTX%PERIODIC state.
```
**// Ethernet 1/1’s PTX state machine in MSTI 2 was moved to the PERIODIC state.**
```
*Mar 18 14:28:41:741 2010 Sysname MSTP/7/MEXS:Slot=1;Instance 2's port105(Ethernet1/1) is selected as MASTER role

**// Ethernet 1/1 was selected as the master port in MSTI 2.**

# In PVST mode, enable STP FSM debugging for all ports in VLAN 2.
```
<Sysname> debugging stp fsm vlan 2
```
*Mar 18 14:28:41:739 2010 Sysname MSTP/7/MEXS:Slot=1;VLAN 2's port105(Ethernet1/1) enters PTX%PERIODIC state.

**// Ethernet 1/1’s PTX state machine on VLAN 2 was moved to the PERIODIC state.**
```
*Mar 18 14:28:41:739 2010 Sysname MSTP/7/MEXS:Slot=1;VLAN 2's port105(Ethernet1/1) enters PTX%MSTP state.
```
**// Ethernet 1/1’s PTX state machine on VLAN 2 was moved to the MSTP state.**
```
*Mar 18 14:28:41:741 2010 Sysname MSTP/7/MEXS:Slot=1;VLAN 2's port105(Ethernet1/1) enters PTX%PERIODIC state.
```
**// Ethernet 1/1’s PTX state machine on VLAN 2 was moved to the PERIODIC state.**
```
*Mar 18 14:28:41:741 2010 Sysname MSTP/7/MEXS:Slot=1;VLAN 2's port105(Ethernet1/1) is selected as MASTER role

**// Ethernet 1/1 was selected as the master port on VLAN 2.**

### debugging stp global-error

Use **debugging stp global-error** to enable STP global error debugging.

Use **undo debugging stp global-error** to disable STP global error debugging.
Syntax

debugging stp global-error
undo debugging stp global-error

Default

STP global error debugging is disabled.

Views

User view

Default command level

2: System level

Usage guidelines

Table 3 describes output fields and messages for the **debugging stp global-error** command.

**Table 207 Output from the debugging stp global-error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port PortID(PortName) received BPDU packet is error for String</td>
<td>Port PortID(PortName) received an error BPDU. The String argument specifies the reason for the error.</td>
</tr>
<tr>
<td>Port PortID(PortName) received error BPDU and discarded it</td>
<td>Port PortID(PortName) received and dropped an error BPDU.</td>
</tr>
<tr>
<td>The protocol type ID is wrong</td>
<td>Incorrect protocol type ID.</td>
</tr>
</tbody>
</table>
| The protocol version ID is wrong | Incorrect protocol version ID. Correct protocol version ID:  
  - 0—STP.  
  - 2—RSTP.  
  - 3—MSTP. |
| Creating mbuffer failed | MSTP failed to create the buffer. |
| P/V semaphore error | Semaphore operation failed. |
| Set STP String error | MSTP failed to set the STP state to the driver. The String argument specifies the STP state, which is **enable** or **disable**. |
| Creating responsive message for configuration failed | MSTP failed to create a configuration response message. |
| Port PortID is inexistent | Port PortID does not exist. |
| Active region configuration error | MSTP failed to activate region configuration. |
| Set instance InstanceID’s port PortID(PortName) STP state String error | The state String of port PortID(PortName) on MSTI InstanceID or VLAN VLANID was set incorrectly. The possible values of the String argument are:  
  - **discarding**.  
  - **learning**.  
  - **forwarding**. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write queue</td>
<td>MSTP failed to write a queue.</td>
</tr>
<tr>
<td><strong>String error</strong></td>
<td>The possible values of the <strong>String</strong> argument are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>MstpMsgQue</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>MstpPktQue</strong>.</td>
</tr>
<tr>
<td>Write event</td>
<td>MSTP failed to write an event.</td>
</tr>
<tr>
<td><strong>String error</strong></td>
<td>The possible values of the <strong>String</strong> argument are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>MSTP_BPDU_EVENT</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>MSTP_L2INF_EVENT</strong>.</td>
</tr>
<tr>
<td>Bind vlan to instance <strong>InstanceID</strong> error</td>
<td>An error occurred when VLANs were bound to MSTI <strong>InstanceID</strong>.</td>
</tr>
<tr>
<td>Instance <strong>InstanceID</strong>'s port <strong>PortID(PortName)</strong> enter unknown state of <strong>String</strong> state machine</td>
<td>The state machine <strong>String</strong> of port <strong>PortID(PortName)</strong> on MSTI <strong>InstanceID</strong> or VLAN VLANID entered an unknown state.</td>
</tr>
<tr>
<td>VLAN VLANID' s port <strong>PortID(PortName)</strong> enter unknown state of <strong>String</strong> state machine</td>
<td>The possible values of the <strong>String</strong> argument are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>PIM</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>PPM</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>TCM</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>PRT_DAB</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>PRT_RDM</strong>.</td>
</tr>
<tr>
<td>Port <strong>PortID(PortName)</strong> send packet error</td>
<td>Port <strong>PortID(PortName)</strong> failed to send packets.</td>
</tr>
</tbody>
</table>

**Examples**

```
# Enable STP global error debugging.
<Sysname> debugging stp global-error
*Mar 18 14:28:41:744 2010 Sysname MSTP/7/PKTERR:Port 10(Ethernet1/1): received BPDU packet is error for FwdDelay Invalid
  // Ethernet 1/1 received an error BPDU. The error was caused by invalid FwdDelay information in the BPDU.
*Mar 18 14:28:41:746 2010 Sysname MSTP/7/PCKERRPORT:Port 10(Ethernet1/1) received error BPDU and discarded it.
  // Ethernet 1/1 received and discarded an error BPDU.
```

debugging stp global-event

Use **debugging stp global-event** to enable STP global event debugging.

Use **undo debugging stp global-event** to disable STP global event debugging.

**Syntax**

```
debugging stp global-event
undo debugging stp global-event
```

**Default**

STP global event debugging is disabled.

**Views**

User view
Default command level

2: System level

Usage guidelines

Table 4 describes output fields and messages for the `debugging stp global-event` command.

### Table 208: Output from the debugging stp global-event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID</td>
<td>Enters PRS Machine</td>
</tr>
<tr>
<td>VLAN ID</td>
<td>Enters PRS Machine</td>
</tr>
<tr>
<td>MSTI Instance ID or VLAN VLAN ID</td>
<td>Enters PRS state machine.</td>
</tr>
</tbody>
</table>

Examples

```
# In MSTP mode, enable STP global event debugging.
<Sysname> debugging stp global-event
*Mar 18 14:28:41:777 2010 Sysname IFNET/5/LINK UPDOWN:
  Ethernet1/1: link status is UP
  // MSTI 0 entered the PRS state machine.
  // MSTI 2 entered the PRS state machine.
# In PVST mode, enable STP global event debugging.
<Sysname> debugging stp global-event
*Mar 18 14:28:41:777 2010 Sysname IFNET/5/LINK UPDOWN:
  Ethernet1/1: link status is UP
  // VLAN 1 entered the PRS state machine.
  // VLAN 2 entered the PRS state machine.
```

debugging stp packet

Use `debugging stp packet` to enable STP BPDU debugging.

Use `undo debugging stp packet` to disable STP BPDU debugging.

**Syntax**

```
debugging stp packet [ receive | send ] [ vlan vlan-id ] [ interface interface-type interface-number ] [ brief | verbose ]
```

```
undo debugging stp packet [ receive | send ] [ vlan vlan-id ] [ interface interface-type interface-number ] [ brief | verbose ]
```

**Default**

STP BPDU debugging is disabled.

**Views**

User view
Default command level

2: System level

Parameters

receive: Specifies debugging for received STP BPDUs.
send: Specifies debugging for sent STP BPDUs.
interface interface-type interface-number: Specifies a port by its type and number. If you do not specify a port, this command enables STP BPDU debugging for all ports.
 vlan vlan-id: Specifies a VLAN by its ID. The value range for the vlan-id argument varies with device models. If you do not specify a VLAN, this command enables STP BPDU debugging for all VLANs. This option is applicable only to the PVST mode.
 brief: Specifies brief STP BPDU debugging.
 verbose: Specifies detailed STP BPDU debugging.

Usage guidelines

If you do not specify the receive and send keywords, this command enables or disables STP BPDU debugging for both received and sent STP BPDUs.

If you do not specify the brief and verbose keywords, this command enables or disables brief STP BPDU debugging.

Table 5 describes output fields and messages for the debugging stp packet command.

Table 209 Output from the debugging stp packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port PortID(PortName) Send Type Packet(Length:Number)</td>
<td>Port PortID(PortName) sent a BPDU of type Type and at length Number (in bytes). The possible values of the Type argument are: • Stp. • Rstp. • Mstp-dot1s. • Mstp-legacy.</td>
</tr>
<tr>
<td>Port PortID(PortName) Rcvd Type Packet(Length:Number)</td>
<td>Port PortID(PortName) received a BPDU of type Type and at length Number (in bytes). The possible values of the Type argument are: • Stp. • Rstp. • Mstp-dot1s. • Mstp-legacy.</td>
</tr>
<tr>
<td>Port PortID(PortName) VLAN VLANID Send Type Packet(Length:Number)</td>
<td>Port PortID(PortName) sent a BPDU of type Type on VLAN VLANID and at length Number (in bytes). The possible values of the Type argument are: • Stp. • Rstp. • Mstp-dot1s. • Mstp-legacy.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Port PortID(PortName) VLAN VLANID Rcvd Type Packet(Length: Number)</td>
<td>Port PortID(PortName) received a BPDU of type Type on VLAN VLANID and at length Number (in bytes). The possible values of the Type argument are: • Stp. • Rstp. • Mstp-dot1s. • Mstp-legacy.</td>
</tr>
<tr>
<td>ProtocolVersionID</td>
<td>Protocol version ID.</td>
</tr>
<tr>
<td>BPDUType</td>
<td>BPDU type.</td>
</tr>
<tr>
<td>CIST Root ID</td>
<td>CIST root bridge ID.</td>
</tr>
<tr>
<td>External RPC</td>
<td>External root path cost.</td>
</tr>
<tr>
<td>Reg Root ID</td>
<td>Regional root bridge ID.</td>
</tr>
<tr>
<td>Internal RPC</td>
<td>Internal root path cost.</td>
</tr>
<tr>
<td>(Instance)Flags: (InstanceID)Port-Role[Flag]</td>
<td>Information about the BPDU: • InstanceID — MSTI ID. • Port-Role — The role of the port that sent or received the BPDU on the named MSTI. The possible port roles are: o Mast — Master port. o Alt — Alternate or backup port. o Root — Root port. o Desi — Designated port. • Flag — BPDU type or port state: o BPDU types include Ta (TCA BPDU), P (proposal BPDU), A (agreement BPDU), and Tc (TC BPDU). o Port states include F (forwarding) and L (learning). If no port state is displayed, the port is in discarding state.</td>
</tr>
<tr>
<td>PKT</td>
<td>Packet debugging information, including: • Port number. • Port name. • Packet direction (inbound/outbound). • Packet type. • Packet length. • Entire packet content displayed in hexadecimal format.</td>
</tr>
</tbody>
</table>

**Examples**

# In MSTP mode, enable debugging for received STP BPDUs on all ports.

```
<Sysname> debugging stp packet receive
```
Internal RPC : 0
CIST Bridge ID : 32768.000f-e200-3700
CIST Port ID : 128.002
(Instance)Flags : (00)Desi[ A P ]

// Ethernet 1/1 received a 103-byte MSTP legacy-format packet.

# In MSTP mode, enable detailed STP BPDU debugging on all ports.
<Sysname> debugging stp packet send verbose
*Mar 18 14:28:41:782 2010 Sysname MSTP/7/PKT:
Port385(Ethernet1/1) Send Mstp-legacy Packet(Length: 103)
00 00 03 02 6c 80 00 00 e0 fc 00 00 00 00 00 00
00 80 00 00 e0 fc 00 00 00 81 81 00 00 14 00 02
00 0f 00 00 00 00 40 30 30 65 30 66 63 30 30 30
30 30 30 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 ac 36 17 7f 50 28 3c
d4 b8 38 21 d8 ab 26 de 62 80 00 00 e0 fc 00 00
00 00 00 00 14 00

// Ethernet 1/1 sent a 103-byte MSTP legacy-format BPDU. The entire 103 bytes of data are displayed.

# In PVST mode, enable debugging for received STP BPDUs on all ports.
<Sysname> debugging stp packet receive
*Mar 18 14:28:41:781 2010 DeviceB MSTP/7/PKT:
Port386(Ethernet1/1) VLAN 2 Rcvd Mstp-legacy Packet(Length: 42)
ProtocolVersionID: 03
BPDUType : 02
Flags : Desi[ P ]

// Ethernet 1/1 received a 42-byte PVST packet.

# In PVST mode, enable detailed STP BPDU debugging on all ports.
<Sysname> debugging stp packet send verbose
*Mar 18 14:28:41:782 2010 DeviceA MSTP/7/PKT:
Port385(Ethernet1/1) VLAN 2 Send Rstp Packet(Length: 42)
00 00 02 02 6c 80 00 00 e0 fc 00 00 00 00 00 00
00 80 00 00 e0 fc 00 00 00 81 81 00 00 14 00 02
00 0f 00 00 00 00 00 00 00 00 00 00 00 00 02

// Ethernet 1/1 sent a 42-byte PVST BPDU. The entire 42 bytes of data are displayed.

debugging stp roles

Use debugging stp roles to enable STP port role change debugging.
Use undo debugging stp roles to disable STP port role change debugging.

Syntax
debugging stp roles
undo debugging stp roles

Default
STP port role change debugging is disabled.
Views
User view

Default command level
2: System level

Usage guidelines

Table 6 describes output fields and messages for the `debugging stp roles` command.

**Table 210 Output from the debugging stp roles command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance InstanceID’s port PortID(PortName) is currently String port</td>
<td>The role of port PortID(PortName) on MSTI InstanceID or VLAN VLANID is String, which can be designated, root, alternate, backup, master, or alternate_boundary.</td>
</tr>
<tr>
<td>VLAN VLANID’s port PortID(PortName) is currently String port</td>
<td></td>
</tr>
</tbody>
</table>

Examples

```
# In MSTP mode, enable STP port role change debugging on the device.
<Sysname> debugging stp roles
*Mar 18 14:28:41:783 2010 Sysname MSTP/7/UPDTROLES:slot=1; The role of ports on instance 2 was updated...
*Mar 18 14:28:41:783 2010 Sysname MSTP/7/ROLES: slot=1;Instance 2's Port106(Ethernet1/1) is currently root port.
// Ethernet 1/1 on MSTI 2 was updated to a root port.
```

```
# In PVST mode, enable STP port role change debugging on the device.
<Sysname> debugging stp roles
*Mar 18 14:28:41:783 2010 Sysname MSTP/7/UPDTROLES:slot=1; The role of ports on VLAN 2 was updated...
*Mar 18 14:28:41:783 2010 Sysname MSTP/7/ROLES: slot=1;VLAN 2's Port106(Ethernet1/1) is currently root port.
// Ethernet 1/1 on VLAN 2 was updated to a root port.
```

debugging stp tc

Use **debugging stp tc** to enable STP TC event debugging.

Use **undo debugging stp tc** to disable STP TC event debugging.

**Syntax**

```
debugging stp tc [ interface interface-type interface-number ]
undo debugging stp tc [ interface interface-type interface-number ]
```

**Default**

STP TC event debugging is disabled.

**Views**

User view
Default command level

2: System level

Parameters

`interface interface-type interface-number`: Specifies a port by its type and number. If you do not specify a port, this command enables STP TC event debugging for all ports.

Usage guidelines

Table 7 describes output fields and messages for the `debugging stp tc` command.

### Table 211 Output from the debugging stp tc command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port PortID(PortName)</td>
<td>TC event <code>String</code> occurred on port <code>PortID(PortName)</code>. The possible values of the <code>String</code> argument are:</td>
</tr>
<tr>
<td></td>
<td>• Receiving TCN—The port received a TCN BPDU.</td>
</tr>
<tr>
<td></td>
<td>• Sending TC—The port sent a TC BPDU.</td>
</tr>
<tr>
<td></td>
<td>• Sending TCA—The port sent a TCA BPDU.</td>
</tr>
<tr>
<td></td>
<td>• Sending TCN—The port sent a TCN BPDU.</td>
</tr>
<tr>
<td>Instance InstanceID’s port PortID(PortName) String</td>
<td>TC event <code>String</code> occurred on port <code>PortID(PortName)</code> in MSTI <code>InstanceID</code> or VLAN <code>VLANID</code>. The possible values of the <code>String</code> argument are:</td>
</tr>
<tr>
<td></td>
<td>• Receiving TC in bound—The port received a TC BPDU on the MSTI.</td>
</tr>
<tr>
<td></td>
<td>• Receiving TCA—The port received a TCA BPDU.</td>
</tr>
<tr>
<td></td>
<td>• Receiving TC—The port received a TC BPDU.</td>
</tr>
<tr>
<td></td>
<td>• Sending TC—The port sent a TC BPDU.</td>
</tr>
<tr>
<td></td>
<td>• Sending TCA—The port sent a TCA BPDU.</td>
</tr>
<tr>
<td></td>
<td>• TcWhile Expiring—TC BPDU transmit timer expired.</td>
</tr>
</tbody>
</table>

Examples

# In MSTP mode, enable STP TC event debugging on all ports.

```
<Sysname> debugging stp tc
*Mar 18 14:28:41:784 2010 Sysname MSTP/8/PORTMSTTC:slot=1; Instance 1's Port106 (Ethernet1/1) Sending TC.
// Ethernet 1/1 on MSTI 1 sent a TC BPDU.
*Mar 18 14:28:41:784 2010 Sysname MSTP/8/PORTMSTTC: slot=1; Instance 1's Port106 (Ethernet1/1) Receiving TC.
// Ethernet 1/1 on MSTI 1 received a TC BPDU.
```

# In PVST mode, enable STP TC event debugging on all ports.

```
<Sysname> debugging stp tc
*Mar 18 14:28:41:784 2010 Sysname MSTP/8/PORTMSTTC:slot=1; VLAN 1's Port106 (Ethernet1/1) Sending TC.
// Ethernet 1/1 in VLAN 1 sent a TC BPDU.
*Mar 18 14:28:41:784 2010 Sysname MSTP/8/PORTMSTTC: slot=1; VLAN 1's Port106 (Ethernet1/1) Receiving TC.
// Ethernet 1/1 in VLAN 1 received a TC BPDU.
```
SSH debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging ssh server

Use `debugging ssh server` to enable SSH server debugging.

Use `undo debugging ssh server` to disable SSH server debugging.

Syntax

```
debugging ssh server { all | vty vty-num { all | error | event | message } }
undo debugging ssh server { all | vty vty-num { all | error | event | message } }
```

Default

SSH server debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- **all**: Specifies all SSH channels.
- **vty-num**: Specifies an SSH channel to be debugged. Its value depends on VTY user interface number and its value range is 0 to 4.
- **all**: Specifies all types of SSH server debugging.
- **error**: Specifies error debugging.
- **event**: Specifies event debugging.
- **message**: Specifies message debugging.

Usage guidelines

Table 1 describes output fields and messages for the `debugging ssh server vty error` command.

**Table 212 Output from the debugging ssh server vty error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>VTY[vty-number]</code></td>
<td>Current user interface.</td>
</tr>
<tr>
<td>Receive error msgtype:</td>
<td>Incorrect message type.</td>
</tr>
<tr>
<td>Process AuthPK Error:</td>
<td>Server public key authentication error.</td>
</tr>
<tr>
<td>Process Password Error:</td>
<td>Password authentication error.</td>
</tr>
<tr>
<td>User Auth Init Error:</td>
<td>Server authentication initialization error.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Service Auth Error:</td>
<td>Identity authentication service request error.</td>
</tr>
<tr>
<td>NewKey Error:</td>
<td>Server newkey state machine error.</td>
</tr>
<tr>
<td>GEX_Reply Error:</td>
<td>GEX public key reply error.</td>
</tr>
<tr>
<td>GEX_Group Error:</td>
<td>Key exchange algorithm error.</td>
</tr>
<tr>
<td>GRP Reply Error:</td>
<td>GRP public key reply error.</td>
</tr>
<tr>
<td>Server Key Init Error:</td>
<td>Algorithm negotiation initialization error.</td>
</tr>
<tr>
<td>SFTPS Opendir:</td>
<td>SFTP server directory open error.</td>
</tr>
<tr>
<td>SFTPS Open Error:</td>
<td>File open error.</td>
</tr>
<tr>
<td>SFTPS Process Error:</td>
<td>SFTP server message process error.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the **debugging ssh server vty event** command.

**Table 213 Output from the debugging ssh server vty event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTY[vty-number]</td>
<td>Current user interface.</td>
</tr>
<tr>
<td>Accept:</td>
<td>Socket accept event.</td>
</tr>
<tr>
<td>Successful to send version string: version-string</td>
<td>Successfully send version information in the format &quot;SSH-&lt;Primary protocol version number&gt;-&lt;Secondary protocol version number&gt;-&lt;Software version number&gt;&quot;</td>
</tr>
<tr>
<td>Socket:socketid</td>
<td>Current socket ID.</td>
</tr>
<tr>
<td>LineIndex:lineindex</td>
<td>Current line index.</td>
</tr>
<tr>
<td>IP:ipaddress</td>
<td>IP address of the login user.</td>
</tr>
<tr>
<td>FSM Change: From fsm1 to fsm2</td>
<td>State machine transition.</td>
</tr>
<tr>
<td>Read:</td>
<td>Socket read event.</td>
</tr>
<tr>
<td>Server Key Init:</td>
<td>Server key initialization.</td>
</tr>
<tr>
<td>InEncrypt: key-algorithm</td>
<td>Received encryption algorithm information.</td>
</tr>
<tr>
<td>OutEncrypt: key-algorithm</td>
<td>Sent encryption algorithm information.</td>
</tr>
<tr>
<td>InMac: mac-algorithm</td>
<td>Received MAC algorithm information.</td>
</tr>
<tr>
<td>OutMac: mac-algorithm</td>
<td>Sent MAC algorithm information.</td>
</tr>
<tr>
<td>KeyType: key-type</td>
<td>Key type.</td>
</tr>
<tr>
<td>SUB1_FSM Change: From sub1 fsm1 to sub1 fsm2</td>
<td>Level-1 sub-state-machine change.</td>
</tr>
<tr>
<td>SUB2_Auth_FSM from sub2 fsm1 to sub2 fsm2</td>
<td>Level-2 sub-state-machine change.</td>
</tr>
<tr>
<td>UserAuthInit:</td>
<td>User authentication initialization.</td>
</tr>
<tr>
<td>Get user name: user-name!</td>
<td>Client username.</td>
</tr>
<tr>
<td>Sub2Password:</td>
<td>Password authentication.</td>
</tr>
<tr>
<td>Channel Request:</td>
<td>Channel request from the client.</td>
</tr>
<tr>
<td>Received channel request: request-type</td>
<td>Type of the received channel request.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STELS Start Shell:</td>
<td>Start shell.</td>
</tr>
<tr>
<td>Receive message:</td>
<td>Received message from SFTP client.</td>
</tr>
<tr>
<td>Successful to create task: Id=taskid</td>
<td>The SSH module successfully created an SFTP task.</td>
</tr>
<tr>
<td>SFTPS Message:</td>
<td>SFTP message type.</td>
</tr>
<tr>
<td>SFTP Server Init:</td>
<td>SFTP version negotiation initialization.</td>
</tr>
<tr>
<td>SFTPS Open:</td>
<td>Opened file.</td>
</tr>
<tr>
<td>SFTPS Close:</td>
<td>Closed file.</td>
</tr>
<tr>
<td>SFTPS Read:</td>
<td>Read file.</td>
</tr>
<tr>
<td>SFTPS Write:</td>
<td>Written file.</td>
</tr>
<tr>
<td>SFTPS Opendir:</td>
<td>Opened directory.</td>
</tr>
<tr>
<td>Readdir:</td>
<td>Read directory.</td>
</tr>
<tr>
<td>UserNameDazzle:</td>
<td>Fake username.</td>
</tr>
<tr>
<td>Session Key Store:</td>
<td>Stored session key.</td>
</tr>
</tbody>
</table>

Table 3 describes output fields and messages for the **debugging ssh server vty message** command.

**Table 214 Output from the debugging ssh server vty message command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTY[vty-number]</td>
<td>Current user interface.</td>
</tr>
<tr>
<td>Send Message:</td>
<td>Sent message to the client.</td>
</tr>
<tr>
<td>SSH_VERSION_SEND from SocketID socketid</td>
<td>Server end sent version information.</td>
</tr>
<tr>
<td>Received:</td>
<td>Received client version.</td>
</tr>
<tr>
<td>Received VersionString: version-string</td>
<td>Received version string.</td>
</tr>
<tr>
<td>SendKexInit:</td>
<td>Sent key negotiation information of server.</td>
</tr>
<tr>
<td>Received Message[Type=type-number]:message-type</td>
<td>Type of the received message.</td>
</tr>
<tr>
<td>GEX_Group:</td>
<td>GEX key exchange algorithm.</td>
</tr>
<tr>
<td>GEX Reply:</td>
<td>GEX key exchange algorithm reply from server.</td>
</tr>
<tr>
<td>Process Channel:</td>
<td>Processed channel message.</td>
</tr>
<tr>
<td>STELS Request PTY:</td>
<td>Stelnet client PTY request.</td>
</tr>
<tr>
<td>STELS Start Shell:</td>
<td>Start shell.</td>
</tr>
<tr>
<td>Process Session:</td>
<td>Processed session message.</td>
</tr>
<tr>
<td>STELS Data:</td>
<td>Processed data message.</td>
</tr>
<tr>
<td>SFTPS Trans:</td>
<td>Sent packet to the client.</td>
</tr>
<tr>
<td>Data Flow Control:</td>
<td>Data flow control at SFTP server side.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable SSH server event debugging for VTY1. When an SSH client logs in to the SSH server, output similar to the following example is generated:
<Sysname> debugging ssh server vty 1 event
*Oct 12 09:32:58:462 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Accept:
Socket:6 LineIndex:83,IP:10.1.1.1

// The client logged in from VTY 1, and the server created a socket.
*Oct 12 09:32:58:463 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Accept:
SSH user comes from 10.1.1.1, and current FSM is SSH_Main_Connect

// The TCP connection was established.
*Oct 12 09:32:58:463 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Send Version To Client:
Successful to send version string: SSH-1.99-Comware-3.4

// The SSH module sent version information to the client.
*Oct 12 09:32:58:464 2006 Sysname SSH/7/Server_EVENT: VTY[1]:FSM Change:
From SSH_Main_Connect to SSH_Main_VersionMatch.

// The client and the server negotiated the version.
*Oct 12 09:32:58:467 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
Process user [] from 10.1.1.1.

// The SSH module processed user data from 10.1.1.1.
*Oct 12 09:32:58:467 2006 Sysname SSH/7/Server_EVENT: VTY[1]:FSM Change:
From SSH_Main_VersionMatch to SSH_Main_SSHProcess.

*Oct 12 09:32:58:564 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
Process user [] from 10.1.1.1.
*Oct 12 09:32:58:565 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
Receive Packet(len=284).

// The SSH module received a packet whose length is 284 bytes.
*Oct 12 09:32:58:566 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Server Key Init:
InEncrypt:aes128-cbc, OutEncrypt:aes128-cbc
InMac:hmac-sha1-96, OutMac:hmac-sha1-96
KeyType:KEX_DH_GEX_SHA1

// The server key was initialized.
*Oct 12 09:32:58:566 2006 Sysname SSH/7/Server_EVENT: VTY[1]:SUB1_FSM Change:
From SSH_Sub1_KEX_Init to SSH_Sub1_KEX_GEX_Group.

*Oct 12 09:32:58:943 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
Process user [] from 10.1.1.1.
*Oct 12 09:32:58:944 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
Receive Packet(len=20).
*Oct 12 09:32:58:944 2006 Sysname SSH/7/Server_EVENT: VTY[1]:SUB1_FSM Change:
From SSH_Sub1_KEX_GEX_Group to SSH_Sub1_KEX_GEX_Reply.

// The SSH module performed GEX algorithm negotiation.
*Oct 12 09:32:58:955 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
Process user [] from 10.1.1.1.
*Oct 12 09:32:59:263 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
Receive Packet(len=268).

*Oct 12 09:32:59:263 2006 Sysname SSH/7/Server_EVENT: VTY[1]:SUB1_FSM Change:
  From SSH_Sub1_KEX_GEX_Reply to SSH_Sub1_KEX_NewKey.

// The SSH module updated the key.

*Oct 12 09:32:59:507 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
  Process user [] from 10.1.1.1.
*Oct 12 09:32:59:508 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
  Receive Packet(len=12).
*Oct 12 09:32:59:509 2006 Sysname SSH/7/Server_EVENT: VTY[1]:SUB1_FSM Change:
  From SSH_Sub1_KEX_NewKey to SSH_Sub1_Authentication.

// The SSH module authenticated the user.

*Oct 12 09:32:59:605 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
  Process user [] from 10.1.1.1.
*Oct 12 09:32:59:606 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
  Receive Packet(len=28).
*Oct 12 09:32:59:607 2006 Sysname SSH/7/Server_EVENT: VTY[1]:UserAuthInit:
  Get user name: client!

// Authentication was initialized.

*Oct 12 09:32:59:707 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
  Process user [] from 10.1.1.1.
*Oct 12 09:32:59:707 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
  Receive Packet(len=44).
*Oct 12 09:32:59:708 2006 Sysname SSH/7/Server_EVENT: VTY[1]:UserAuthInit:
  Get user name: client!

// The login username is client.

*Oct 12 09:32:59:709 2006 Sysname SSH/7/Server_EVENT: VTY[1]:UserAuthInit:
  Current AuthType is SSH_AUTH_PASSWORD

// The authentication method is password authentication.

*Oct 12 09:33:01:585 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
  Process user [client] from 10.1.1.1.
*Oct 12 09:33:01:585 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
  Receive Packet(len=60).
*Oct 12 09:33:01:586 2006 Sysname SSH/7/Server_EVENT: VTY[1]:UserAuthInit:
  Get user name: client!
*Oct 12 09:33:01:587 2006 Sysname SSH/7/Server_EVENT: VTY[1]:SUB2_Auth_FSM
  from SSH_Sub2_Service_Acc to SSH_Sub2_Auth_Init
*Oct 12 09:33:01:587 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Sub2Password:
  User:client  PasswordLen: 6

// The password length of user client is 6.

*Oct 12 09:33:01:613 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Sub2Password: SSH user client
  succeeded to login from 10.1.1.1 on VTY1.
*Oct 12 09:33:01:614 2006 Sysname SSH/7/Server_EVENT: VTY[1]:LOGIN Succeed:
  SSH user client succeeded to login from 10.1.1.1(000f-e200-0001) on VTY1.

// The user client passed the authentication.
Oct 12 09:33:01:615 2006 Sysname SSH/7/Server_EVENT: VTY[1]:SUB2_Auth_FSM
from SSH_Sub2_Auth_Password to SSH_Sub2_Auth_Init

// Level-2 sub-state-machine returned to the authentication initialization status.

Oct 12 09:33:01:615 2006 Sysname SSH/7/Server_EVENT: VTY[1]:SUB1_FSM Change:
From SSH_Sub1_Authentication to SSH_Sub1_Channel.

// Channel request.

Oct 12 09:33:01:696 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
Process user [client] from 10.1.1.1.

Oct 12 09:33:01:697 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
Receive Packet(len=44).

Oct 12 09:33:01:697 2006 Sysname SSH/7/Server_EVENT: VTY[1]:SUB1_FSM Change:
From SSH_Sub1_Channel to SSH_Sub1_Session.

// The SSH module established a session.

Oct 12 09:33:01:796 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
Process user [client] from 10.1.1.1.

Oct 12 09:33:01:797 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
Receive Packet(len=60).

Oct 12 09:33:01:797 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Channel Request:
Received channel request: pty-req

// The SSH module received a channel request whose type is pty-req.

Oct 12 09:33:01:798 2006 Sysname SSH/7/Server_EVENT: VTY[1]:STELS Request PTY:
Successful to send SSH2_MSG_CHANNEL_SUCCESS(99) from 10.1.1.2 to 10.1.1.1

// The SSH module successfully sent SSH2_MSG_CHANNEL_SUCCESS message.

Oct 12 09:33:01:897 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read:
Process user [client] from 10.1.1.1.

Oct 12 09:33:01:898 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Read Buffer:
Receive Packet(len=28).

Oct 12 09:33:01:898 2006 Sysname SSH/7/Server_EVENT: VTY[1]:Channel Request:
Received channel request: shell

// The SSH module received a channel request whose type is shell.

Oct 12 09:33:01:899 2006 Sysname SSH/7/Server_EVENT: VTY[1]:STELS Start Shell:
Send SSH2_MSG_CHANNEL_SUCCESS(99) from 10.1.1.2 to 10.1.1.1

Oct 12 09:33:02:01 2006 Sysname SHELL/4/LOGIN: client login from 10.1.1.1

// The client logged into the server successfully.

debugging ssh client

Syntax

Use debugging ssh client to enable SSH client debugging.

Use undo debugging ssh client to disable SSH client debugging.
debugging ssh client { all | error | event | message }
undo debugging ssh client { all | error | event | message }

**Default**

SSH client debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

all: Specifies all types of SSH client debugging.
error: Specifies SSH client error debugging.
event: Specifies SSH client event debugging.
message: Specifies SSH client message debugging.

**Usage guidelines**

Table 4 describes output fields and messages for the `debugging ssh client event` command.

**Table 215 Output from the debugging ssh client event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessSession:</td>
<td>Process session.</td>
</tr>
<tr>
<td>InEncrypt: <em>key-algorithm</em></td>
<td>Received encryption algorithm information.</td>
</tr>
<tr>
<td>OutEncrypt: <em>key-algorithm</em></td>
<td>Sent encryption algorithm information.</td>
</tr>
<tr>
<td>InMac: <em>mac-algorithm</em></td>
<td>Received MAC algorithm information.</td>
</tr>
<tr>
<td>OutMac: <em>mac-algorithm</em></td>
<td>Sent MAC algorithm information.</td>
</tr>
<tr>
<td>Process Kex Init:</td>
<td>Algorithm negotiation initialization.</td>
</tr>
<tr>
<td>Connect Socket:</td>
<td>Socket connection.</td>
</tr>
<tr>
<td>FSM from fsm1 to fsm2</td>
<td>State machine changes from connection status to version negotiation.</td>
</tr>
<tr>
<td>Client_SUB1_FSM from fsm1 to fsm2</td>
<td>Client level-1 sub-state-machine change.</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the `debugging ssh client message` command.

**Table 216 Output from the debugging ssh client message command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STELC:</td>
<td>Stelnet client.</td>
</tr>
<tr>
<td>Client Data Flow Control:</td>
<td>Client data flow control.</td>
</tr>
<tr>
<td>Send Disconnect:</td>
<td>Send disconnection message.</td>
</tr>
<tr>
<td>Window Adjust:</td>
<td>Client channel window adjust.</td>
</tr>
<tr>
<td>AuthReq:</td>
<td>Sent authentication request.</td>
</tr>
<tr>
<td>ServiceReq:</td>
<td>Sent service request.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>NewKey:</td>
<td>The client stayed in newkey state.</td>
</tr>
<tr>
<td>GEX Init:</td>
<td>GEX algorithm initialization.</td>
</tr>
<tr>
<td>GEX Request:</td>
<td>GEX request.</td>
</tr>
<tr>
<td>Client_SendVersionString:</td>
<td>The client sent version string.</td>
</tr>
<tr>
<td>SFTPC:</td>
<td>SFTP client.</td>
</tr>
<tr>
<td>SFTPC CUSTOM CLOSED</td>
<td>Client close message.</td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the debugging ssh client error command.

Table 217 Output from the debugging ssh client error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessSession Error:</td>
<td>Session processing error.</td>
</tr>
<tr>
<td>Error:</td>
<td>Error message.</td>
</tr>
<tr>
<td>GEX Init Error:</td>
<td>GEX algorithm initialization error.</td>
</tr>
<tr>
<td>GRP Init Error:</td>
<td>GRP algorithm initialization error.</td>
</tr>
<tr>
<td>Process Kex Init Error:</td>
<td>Key exchange error.</td>
</tr>
<tr>
<td>VersionString Received Error:</td>
<td>Version string receive error.</td>
</tr>
<tr>
<td>DoClose:</td>
<td>File close error.</td>
</tr>
<tr>
<td>Process RealPath Error:</td>
<td>Error occurs when relative path is changed to absolute path.</td>
</tr>
<tr>
<td>Connect Socket Error:</td>
<td>Socket connection error.</td>
</tr>
</tbody>
</table>

Examples

# Enable debugging for SSH client. When an SSH client logs in to the SSH server, output similar to the following example is generated:

```
<Sysname> debugging ssh client event
<Sysname> ssh2 10.1.1.2
Username: client
Trying 10.1.1.2 ...
Press CTRL+K to abort
Connected to 10.1.1.2 ...
*Oct 12 09:21:00:252 2006 Sysname SSH/7/Client_EVENT: FSM from SSH_Main_Connect to SSH_Main_VersionMatch
  // The client and the server negotiated the version.
*Oct 12 09:21:00:254 2006 Sysname SSH/7/Client_EVENT: FSM from SSH_Main_Connect to SSH2_Main_KEX_Init
  // The SSH module received a packet whose length is 284 bytes.
*Oct 12 09:21:00:478 2006 Sysname SSH/7/Client_EVENT: Process Kex Init:
  InEncrypt:aes128-cbc, OutEncrypt:aes128-cbc
  InMac:hmac-sha1-96, OutMac:hmac-sha1-96
```
// The SSH module negotiated the key.
*Oct 12 09:21:00:479 2006 Sysname SSH/7/Client_EVENT: FSM from SSH2_Main_KEX_Init to SSH2_Main_KEX_GEX_Request
*Oct 12 09:21:00:889 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
    Receive Packet(len=276).
*Oct 12 09:21:00:889 2006 Sysname SSH/7/Client_EVENT: FSM from SSH2_Main_KEX_GEX_Request to SSH2_Main_KEX_GEX_Init

// The SSH module performed GEX algorithm negotiation.
*Oct 12 09:21:01:441 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
    Receive Packet(len=572).
*Oct 12 09:21:01:441 2006 Sysname SSH/7/Client_EVENT: FSM from SSH2_Main_KEX_GEX_Init to SSH2_Main_KEX_NewKey

// The SSH module updated the key.
*Oct 12 09:21:01:539 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
    Receive Packet(len=12).
*Oct 12 09:21:01:540 2006 Sysname SSH/7/Client_EVENT: FSM from SSH2_Main_KEX_NewKey to SSH2_Main_Authentication

// The SSH module authenticated the user.
*Oct 12 09:21:01:640 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
    Receive Packet(len=28).
*Oct 12 09:21:01:641 2006 Sysname SSH/7/Client_EVENT: Client_SUB1_FSM from SSH2_Sub1_Service_Req to SSH2_Sub1_Auth_Req
Enter password:

// The SSH module prompted the user to enter password.
*Oct 12 09:21:01:739 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
    Receive Packet(len=28).

*Oct 12 09:21:09:841 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
    Receive Packet(len=12).
*Oct 12 09:21:09:842 2006 Sysname SSH/7/Client_EVENT: Client_SUB1_FSM from SSH2_Sub1_Auth_Req to SSH2_Sub1_Service_Req

// The SSH module generated a service request.
*Oct 12 09:21:09:843 2006 Sysname SSH/7/Client_EVENT: FSM from SSH2_Main_Authentication to SSH2_Main_Channel

// The SSH module generated a channel request.
*Oct 12 09:21:09:941 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
    Receive Packet(len=28).
*Oct 12 09:21:09:942 2006 Sysname SSH/7/Client_EVENT: FSM from SSH2_Main_Channel to SSH2_Main_Pty

// The SSH module sent a channel request whose type is PTY.
*Oct 12 09:21:10:42 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
    Receive Packet(len=12).
*Oct 12 09:21:10:42 2006 Sysname SSH/7/Client_EVENT: FSM from SSH2_Main_Pty to SSH2_Main_Shell

// The SSH module sent a channel request whose type is shell.
Oct 12 09:21:10:141 2006 Sysname SSH/7/Client_EVENT: Read Buffer:
  Receive Packet(len=12).
Oct 12 09:21:10:142 2006 Sysname SSH/7/Client_EVENT: FSM from SSH2_Main_Shell to
  SSH2_Main_Session

// The SSH module established a session.
SSL debugging commands

debugging ssl

Use `debugging ssl` to enable SSL debugging. Use `undo debugging ssl` to disable SSL debugging.

**Syntax**

```
debugging ssl { all | error | event | packet }
undo debugging ssl { all | error | event | packet }
```

**Default**

SSL debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- `all`: Specifies all types of SSL debugging.
- `error`: Specifies SSL error debugging.
- `event`: Specifies SSL event debugging.
- `packet`: Specifies SSL packet debugging.

**Usage guidelines**

If you enable debugging for SSL packets, contents of inbound and outbound SSL packets will be displayed in hexadecimal.

Table 1 and Table 2 describe output fields and messages for the `debugging ssl error` command and the `debugging ssl event` command.

**Table 218 Output from the debugging ssl error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Failed to handle encrypted data, operation type: `operation-type`. | Failed to handle the encrypted data, which may be of the following types:  
  - 20: SSL change cipher spec protocol packet.  
  - 21: SSL alert protocol packet.  
  - 23: SSL record protocol packet. |
| Failed to bind address to SSL. | Failed to bind the IP address to SSL. |
**Table 219 Output from the debugging ssl event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN/OUT</td>
<td>Inbound/outbound SSL protocol message.</td>
</tr>
</tbody>
</table>

**[ length length ], message-type**
Information about the received or sent SSL 2.0 protocol message, including the length of the message (in hexadecimal) and the message type.

Possible message types are:
- CLIENT-HELLO.
- CLIENT-MASTER-KEY.
- CLIENT-FINISHED.
- SERVER-HELLO.
- SERVER-VERIFY.
- SERVER-FINISHED.
- REQUEST-CERTIFICATE.
- CLIENT-CERTIFICATE.
- Unknown.

**[ length length ], ERROR: reason**
SSL 2.0 error information, including the length of the packet in error and the reason.

Possible reasons are:
- NO-CIPHER-ERROR.
- NO-CERTIFICATE-ERROR.
- BAD-CERTIFICATE-ERROR.
- UNSUPPORTED-CERTIFICATE-TYPE-ERROR.
- Unknown.

**Handshake [ length length ] message-type**
Information about the received or sent SSL 3.0 handshake protocol message, including the length of the message (in hexadecimal) and the message type.

Possible message types are:
- HelloRequest.
- ClientHello.
- ServerHello.
- Certificate.
- ServerKeyExchange.
- CertificateRequest.
- ServerHelloDone.
- CertificateVerify.
- ClientKeyExchange.
- Finished.
- Unknown.

**ChangeCipherSpec [ length length ]**
Information about the received or sent SSL 3.0 change cipher spec protocol message, including the length of the message (in hexadecimal).
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Alert [ length length ] level description | Information about the received or sent SSL 3.0 alert protocol message, including the length of the message (in hexadecimal), the error level, and the message description. The error level can be warning or fatal. Possible message descriptions are:  
  • close_notify.  
  • unexpected_message.  
  • bad_record_mac.  
  • decryption_failed.  
  • record_overflow.  
  • decompression_failure.  
  • handshake_failure.  
  • no_certificate.  
  • bad_certificate.  
  • unsupported_certificate.  
  • certificate_revoked.  
  • certificate_expired.  
  • certificate_unknown.  
  • illegal_parameter.  
  • unknown_ca.  
  • access_denied.  
  • decode_error.  
  • decrypt_error.  
  • export_restriction.  
  • protocol_version.  
  • insufficient_security.  
  • internal_error.  
  • user_canceled.  
  • no_renegotiation.  
  • unknown. |
| Valid finished message from peer, session resume flag: flag. | Verified the finished message from the peer successfully. flag is the session resuming flag and can be 0 or 1. 0 means that the session is a newly established one, and 1 means that the session is a resumed one. |

**SSL_CREATESSL**  
Creating the SSL connection.  

**Choose cipher: NULL.**  
Choosing the encryption algorithm: No encryption algorithm available.  

**SSL_accept**  
SSL handshake process.  

**before string initialization.**  
Before SSL or accept initialization. string can be SSL or accept.  

**SSL renegotiate ciphers.**  
SSL is renegotiating the encryption algorithm.  

**before/accept initialization.**  
Initializing the before or accept state.  

**ok/accept SSL initialization.**  
Initializing the ok or accept state.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSLv2 read client</td>
<td>SSLv2 read a client side message, which may be hello A, hello B, hello C, master key A, master key B, finished A, or finished B.</td>
</tr>
<tr>
<td>SSLv2 write client</td>
<td>SSLv2 wrote a client side message, which may be hello A, hello B, master key A, master key B, finished A, finished B, certificate A, certificate B, certificate C, or certificate D.</td>
</tr>
<tr>
<td>SSLv2 write server</td>
<td>SSLv2 wrote a server side message, which may be hello A, hello B, verify A, verify B, verify C, finished A, or finished B.</td>
</tr>
<tr>
<td>SSLv2 write request certificate</td>
<td>SSLv2 wrote a request certificate, which may be A, B, C, or D.</td>
</tr>
<tr>
<td>SSLv3 read server</td>
<td>SSLv3 read a server side message, which may be hello A, hello B, key exchange A, key exchange B, certificate request A, certificate request B, done A, or done B.</td>
</tr>
<tr>
<td>SSLv3 read client</td>
<td>SSLv3 read a client side message, which may be hello A, hello B, hello C, certificate A, certificate B, key exchange A, or key exchange B.</td>
</tr>
<tr>
<td>SSLv3 write client</td>
<td>SSLv3 wrote a client side message, which may be hello A, hello B, certificate A, certificate B, certificate C, certificate D, key exchange A, or key exchange B.</td>
</tr>
<tr>
<td>SSLv3 write server</td>
<td>SSLv3 wrote a server side message, which may be hello A, hello B, done A, or done B.</td>
</tr>
<tr>
<td>SSLv3 write</td>
<td>SSLv3 wrote a message, which may be certificate verify A, certificate verify B, change cipher spec A, change cipher spec B, finished A, finished B, hello request A, hello request B, hello request C, certificate A, certificate B, key exchange A, key exchange B, certificate request A, or certificate request B.</td>
</tr>
<tr>
<td>SSLv3 read</td>
<td>SSLv3 read a message, which may be certificate verify A, certificate verify B, finished A, finished B, change cipher spec A, or change cipher spec B.</td>
</tr>
<tr>
<td>SSLv2/v3 write client</td>
<td>SSLv2/v3 wrote a client side message, which may be hello A or hello B.</td>
</tr>
<tr>
<td>SSLv2/v3 read client</td>
<td>SSLv2/v3 read a client side message, which may be hello A or hello B.</td>
</tr>
<tr>
<td>SSLv2/v3 read server</td>
<td>SSLv2/v3 read a server side message, which may be hello A or hello B.</td>
</tr>
<tr>
<td>SSLv3 flush data</td>
<td>SSLv3 is preparing for the next process.</td>
</tr>
<tr>
<td>SSL_accept end successfully</td>
<td>The SSL handshake process ended successfully.</td>
</tr>
<tr>
<td>SSL_accept end unsuccessfully</td>
<td>The SSL handshake process failed.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable debugging for SSL events on a device configured with HTTPS service. When an SSL client accesses the HTTPS server, output similar to the following example is generated:

```
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging ssl event

// SSL accepted an SSLv3 connection request and sent an acknowledgement message. The SSL ID of the connection is 24.
```
SSL started the handshake process.

SSL was initializing the BEFORE/ACCEPT state.

SSL received a ClientHello handshake packet from the client. The protocol version is SSL 2.0 and the packet length is 76 bytes.

The above shows algorithms that the server and client support.

SSL chose RC4-MD5 as the encryption algorithm.

SSL chose RC4 as the encryption algorithm and MD5 as the digest algorithm.

SSLv3 read a hello A packet from the client.

SSL sent a ServerHello handshake packet to the client. The protocol version is SSL 3.0 and the packet length is 74 bytes.

SSL wrote the hello A packet that the server sent to the client.

SSL sent a Certificate packet. The protocol version is SSL 3.0 and the packet length is 1435 bytes.
SSL sent a ServerHelloDone packet to the client. The protocol version is SSL 3.0 and the packet length is 4 bytes.

SSLv3 wrote the ServerHelloDone packet that the server sent to the client.

SSL received a ClientKeyExchange packet from the client. The protocol version is SSL 3.0 and the packet length is 132 bytes.

SSLv3 read the ClientKeyExchange packet that was received from the client.

SSL received a Finished packet from the client. The protocol version is SSL 3.0 and the packet length is 40 bytes.

SSLv3 read the Finished packet received from the client.

SSL sent a ChangeCipherSpec packet to the client. The protocol version is SSL 3.0 and the packet length is 1 byte.

SSL wrote the ChangeCipherSpec packet that the server sent to the client.

SSL sent a Finished packet to the client. The protocol version is SSL 3.0 and the packet length is 40 bytes.

SSL wrote the Finished packet that the server sent to the client.

SSL accept process finished successfully.

Handshake process finished. The return result 1, indicating that the process is successful.

Invalid message. The SSL ID is 24, the flag bit is 2, and the external control block flag bit is 3.
// SSL was closing the SSL connection. The SSL ID is 24 and the socket ID is 3.

// SSL wrote an alert packet. The alert level is warning and the description is close_notify.
*Nov 9 17:06:19:678 2008 Sysname SSL/7/EVENT:OUT: SSL 3.0 Alert [length 0002], warning close_notify

// SSL sent an alert packet to the client. The protocol version is SSL 3.0, alert level is warning, description is close_notify, and packet length is 2 bytes.
SSL VPN debugging commands

describing ssl-vpn

Use debugging ssl-vpn to enable SSL VPN debugging.
Use undo debugging ssl-vpn to disable SSL VPN debugging.

Syntax

ddebugging ssl-vpn { all | ipac | tlac | wac } { all | error | event | packet }
undo debugging ssl-vpn { all | ipac | tlac | wac } { all | error | event | packet }

Default

SSL VPN debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

all: Specifies debugging for IP network accesses, TCP application accesses, and website accesses.
ipac: Specifies debugging for IP network accesses.
 tlac: Specifies debugging for TCP application accesses.
wac: Specifies debugging for website accesses.
al: Specifies debugging for SSL VPN errors, events, and packets.
error: Specifies debugging for SSL VPN errors.
event: Specifies debugging for SSL VPN events.
packet: Specifies debugging for SSL VPN packets.

Examples

# Enable all types of debugging for website accesses. When a user accesses WAC resources through
the SSL VPN Web interface, output similar to the following example is generated:
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging ssl-vpn wac all
*Jun 5 10:15:09:78 2008 Sysname SVN/7/WAC_EVENT: Resource 0x214800 is authorized to user
0x44b70401.
// Resource 0x214800 was authorized to user 0x44b70401.
*Jun 5 10:15:09:78 2008 Sysname SVN/7/WAC_EVENT: Succeeded in adding web site
// Website http://192.168.1.100/test/ was added to the connection pool.
*Jun 5 10:15:09:78 2008 Sysname SVN/7/WAC_EVENT: Succeeded in creating new connection,
host IP address: 192.168.1.100, port: 20480.
A new connection was created. The IP address of the Web server is 192.168.1.100.

*Jun 5 10:15:09:80 2008 Sysname SVVPN/7/WAC_EVENT: Succeeded in putting connection to pool, site name is http://192.168.1.100/test/.

A connection was added. The website is http://192.168.1.100/test/.


The SSL connection was delivered to the Web server. The connection ID is 327681.

*Jun 5 10:15:09:125 2008 Sysname SVVPN/7/WAC_EVENT: File handler 1 has been deleted.

File handler 1 has been deleted. This message appears if no operation is performed on the SSL connection during a specific period.


Website http://192.168.1.100/test/ was deleted from the connection pool.


A connection was released successfully. The website is http://192.168.1.100/test/.


The SSL connection was delivered to the Web server and then closed. The connection ID is 327681.
Stateful failover debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

The stateful failover module name is identified as "DHBK" in debugging messages.

debugging dhbk

Use `debugging dhbk state` to enable stateful failover state debugging.

Use `undo debugging dhbk state` to disable stateful failover state debugging.

Syntax

```
debugging dhbk state { all | error | event | hello }
undo debugging dhbk state { all | error | event | hello }
```

Default

Stateful failover debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `all`: Specifies all types of stateful failover state debugging.
- `error`: Specifies stateful failover state error debugging.
- `event`: Specifies stateful failover state event debugging.
- `hello`: Specifies stateful failover state debugging for Hello packets.

Usage guidelines

Table 1 describes output fields and messages for the `debugging dhbk state` command.

**Table 220 Output from the debugging dhbk state command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event: event</td>
<td>Stateful failover event notified to the service module:</td>
</tr>
<tr>
<td></td>
<td>• 1—The state switches to <code>Independence</code>.</td>
</tr>
<tr>
<td></td>
<td>• 2—The state switches to <code>Synchronize</code>.</td>
</tr>
<tr>
<td></td>
<td>• 3—The backup type has changed.</td>
</tr>
<tr>
<td>BackupType: backuptype</td>
<td>Backup type:</td>
</tr>
<tr>
<td></td>
<td>• 0—Stateful failover is not enabled and the backup type is invalid.</td>
</tr>
<tr>
<td></td>
<td>• 1—Asymmetric path is not supported.</td>
</tr>
<tr>
<td></td>
<td>• 2—Asymmetric path is supported.</td>
</tr>
<tr>
<td>MID: mid</td>
<td>ID of the service module.</td>
</tr>
</tbody>
</table>
Table 2 describes output fields and messages for the `debugging dhbk state event` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogID: logid</td>
<td>Log ID.</td>
</tr>
<tr>
<td>Version: version</td>
<td>Version information included in Hello packets.</td>
</tr>
</tbody>
</table>

### Table 221 Output from the debugging dhbk state event command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Action: Update DHBK state, current state: `curr-state`; next state: `next-state`. | The hot backup state was updated.  
  - `curr-state`—Current hot backup state.  
  - `next-state`—Hot backup state after update. |
| Send Hello packet too frequently, ignore it. | The Hello packet will not be sent this time because it has been sent too frequently. |

Table 3 describes output fields and messages for the `debugging dhbk state error` command.

### Table 222 Output from the debugging dhbk state error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to append information to message, OID: <code>oid</code>.</td>
<td>DHBK failed to append information to the message. <code>oid</code> refers to object ID.</td>
</tr>
<tr>
<td>DHBK was not initialized when module registered to it, MID: <code>mid</code>.</td>
<td>DHBK had not completed initialization when the service module registered with it.</td>
</tr>
</tbody>
</table>

### Examples

#### # Enable stateful failover state event debugging when the stateful failover enters Synchronize state. Then, enable stateful failover again, and change the backup type.

```bash
<Sysname> debugging dhbk state event
*Aug 6 14:30:55:312 2007 H3C DHBK/4/EVENT: Enable DHBK function when it was enabled, Old BackupType: 2; new BackupType: 1.
  // The backup type was changed from 2 (supports asymmetric path) to 1 (does not support asymmetric path). The log 0x2921005 was printed, and the change to backup type 1 was notified to the registered module.
```

#### # Enable state debugging for Hello packets when stateful failover enters Synchronize state.

```bash
<Sysname> debugging dhbk state hello
  // The received Hello packet had a matching version, and the Hello packet counter was reset. The number of Hello packets before reset was 1.
```
debugging dhbk service

Use `debugging dhbk service` to enable stateful failover service debugging.

Use `undo debugging dhbk service` to disable stateful failover service debugging.

Syntax

```
debugging dhbk service { all | error | event | fsm | info }
undo debugging dhbk service { all | error | event | fsm | info }
```

Default

Stateful failover service debugging is disabled.

Views

User view

Parameters

- **all**: Specifies all types of stateful failover service debugging.
- **error**: Specifies stateful failover service error debugging.
- **event**: Specifies stateful failover service event debugging.
- **fsm**: Specifies stateful failover service FSM debugging.
- **info**: Specifies stateful failover service object information debugging.

Usage guidelines

Table 4 describes output fields and messages for the `debugging dhbk service event` command.

Table 223 Output from the `debugging dhbk service event` command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response: <code>request</code></td>
<td>DHBK responded to the message from the management module:</td>
</tr>
<tr>
<td></td>
<td>• Start sending or receiving backup information.</td>
</tr>
<tr>
<td></td>
<td>• Stop sending or receiving backup information.</td>
</tr>
<tr>
<td>Received: <code>notify</code></td>
<td>DHBK received the message from the management module:</td>
</tr>
<tr>
<td></td>
<td>• Synchronous event.</td>
</tr>
<tr>
<td></td>
<td>• Non-synchronous event.</td>
</tr>
<tr>
<td>Backup objects:</td>
<td>Backup objects:</td>
</tr>
<tr>
<td></td>
<td>• Common session.</td>
</tr>
<tr>
<td></td>
<td>• Father session.</td>
</tr>
<tr>
<td></td>
<td>• Force session.</td>
</tr>
<tr>
<td></td>
<td>• Session relation-table.</td>
</tr>
<tr>
<td></td>
<td>• NAT attached information.</td>
</tr>
<tr>
<td></td>
<td>• ALG attached information.</td>
</tr>
<tr>
<td></td>
<td>• Blacklist.</td>
</tr>
<tr>
<td></td>
<td>• Aging message.</td>
</tr>
<tr>
<td></td>
<td>• LB (load balancing) session.</td>
</tr>
<tr>
<td></td>
<td>• LB persistence entry backup succeeded.</td>
</tr>
<tr>
<td></td>
<td>• Received synchronization event when backing up LB session.</td>
</tr>
<tr>
<td></td>
<td>• Received stop-synchronization event when backing up LB session.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Restore objects: restore-object</td>
<td></td>
</tr>
<tr>
<td>Restored objects:</td>
<td></td>
</tr>
<tr>
<td>• Common session.</td>
<td></td>
</tr>
<tr>
<td>• Father session.</td>
<td></td>
</tr>
<tr>
<td>• Force session.</td>
<td></td>
</tr>
<tr>
<td>• Session relation-table.</td>
<td></td>
</tr>
<tr>
<td>• NAT attached information.</td>
<td>The attached information will be backed up as the appended data of the session backup data</td>
</tr>
<tr>
<td>• ALG attached information.</td>
<td></td>
</tr>
<tr>
<td>• Blacklist.</td>
<td></td>
</tr>
<tr>
<td>• Aging message.</td>
<td></td>
</tr>
<tr>
<td>• LB session.</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages for the **debugging dhbk service error** command.

**Table 224 Output from the debugging dhbk service error command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting status failed</td>
<td></td>
</tr>
<tr>
<td>DHBK failed to set the following status:</td>
<td></td>
</tr>
<tr>
<td>• Session status.</td>
<td></td>
</tr>
<tr>
<td>• Blacklist status.</td>
<td></td>
</tr>
<tr>
<td>• Session debug status.</td>
<td></td>
</tr>
<tr>
<td>• Blacklist debug status.</td>
<td></td>
</tr>
</tbody>
</table>

| Backing up information failed  |             |
| DHBK failed to back up the following object: |             |
| • Common session.              |             |
| • Father session.              |             |
| • Attached information of session. |             |
| • Attached information of force session. |             |
| • Blacklist.                   |             |
| • NAT attached information.    |             |
| • ALG attached information.    |             |
| • FLT attached information.    |             |

| Restore information failed     |             |
| DHBK failed to restore the following backup data: |             |
| • Common session.              |             |
| • Father session.              |             |
| • Attached information of session. |             |
| • Attached information of force session. |             |
| • Blacklist.                   |             |
| • NAT attached information.    |             |
| • ALG attached information.    |             |
| • FLT attached information.    |             |

| LB backup: Failed to get real service name by ID. (IP: IP-address, Port: Port) | DHBK failed to get real service name by ID during LB data backup: |
| IP—IP address of the real service. |             |
| Port—Port number of the real service. |             |
Table 6 describes output fields and messages for the debugging dhbk service fsm command.

Table 225 Output from the debugging dhbk service fsm command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| LB restore: Failed to get virtual service. (VPN ID: VPN-ID, IP: IP-address, Port: Port, Protocol: Protocol) | DHBK failed to get virtual service during LB data restoration:  
  • **VPN-ID**—VPN ID.  
  • **IP-address**—IP address of the virtual service.  
  • **Port**—Port number of the virtual service.  
  • **Protocol**—Protocol type of the virtual service. |
| LB backup: MBuffer insufficient. | DHBK failed to find valid MBUF during LB data backup because of insufficient memory. |

Table 7 describes output fields and messages for the debugging dhbk service info command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Aging information: aging-info | Events that can trigger aging:  
  • Timer triggered.  
  • Received aging request.  
  • Received aging response.  
  • Received unable aging response.  
  • Received compulsive aging message.  
  • Received launch aging packet.  
  • Received refresh aging time packet.  
  • Received compulsive aging command.  
  • Age session. |
| ReqTuple6: srcIP/ srcPort->destIP/ destPort (ProtoType) VPN: VPN | Request 6-tuple:  
  Source IP address/source port number—>destination IP address/destination port number (protocol type) VPN index. |
| RespTuple6: srcIP/ srcPort->destIP/ destPort (ProtoType) VPN: VPN | Response 6-tuple:  
  Source IP address/source port number—>destination IP address/destination port number (protocol type) VPN index. |
| FSM: preQueue ->nextQueue | The session enters the nextQueue from the preQueue. |
Table 226 Output from the debugging dhbk service info command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT attached information:</td>
<td>• Local IP/Port—Private IP address and port number.</td>
</tr>
<tr>
<td>Local IP/Port: Local-IP/Port</td>
<td>• Global IP/Port—Public IP address and port number.</td>
</tr>
<tr>
<td>Global IP/Port: Global-IP/Port</td>
<td>• PatType—PAT type, including PAT, NOPAT, STATIC (one-to-one NAT), and SERVER (NAT internal server).</td>
</tr>
<tr>
<td>PatType: PatType</td>
<td>• VPN—VPN index.</td>
</tr>
<tr>
<td>VPN: VPN</td>
<td></td>
</tr>
<tr>
<td>ALG attached information:</td>
<td>• FSM—FSM state.</td>
</tr>
<tr>
<td>FSM: State</td>
<td>• Seq—Request message.</td>
</tr>
<tr>
<td>Seq: INIT: number</td>
<td>• Ack—Reply message.</td>
</tr>
<tr>
<td>Adjust: number</td>
<td>• INIT—Initial sequence number.</td>
</tr>
<tr>
<td>Current: number</td>
<td>• Adjust—Adjusted sequence number.</td>
</tr>
<tr>
<td>Now: number</td>
<td>• Current—Sequence number before adjustment.</td>
</tr>
<tr>
<td>Ack: INIT: number</td>
<td>• Now—Sequence number after adjustment.</td>
</tr>
<tr>
<td>Adjust: number</td>
<td>• NAT—Address translation information.</td>
</tr>
<tr>
<td>Current: number</td>
<td>• Original IP/Port—Original IP address/port number.</td>
</tr>
<tr>
<td>Now: number</td>
<td>• New IP/Port—I P address/port number after translation.</td>
</tr>
<tr>
<td>Ack: INIT: number</td>
<td>• PatType—PAT type, including PAT, NOPAT, STATIC (one-to-one NAT), and SERVER (NAT internal server).</td>
</tr>
<tr>
<td>Adjust: number</td>
<td></td>
</tr>
<tr>
<td>NAT: Original IP/Port: Original IP/Port</td>
<td></td>
</tr>
<tr>
<td>New IP/Port: New IP/Port</td>
<td></td>
</tr>
<tr>
<td>PatType: PatType</td>
<td></td>
</tr>
<tr>
<td>Relation-table information:</td>
<td></td>
</tr>
<tr>
<td>Local IP/Port: Local-IP/Port</td>
<td></td>
</tr>
<tr>
<td>Global IP/Port: Global-IP/Port</td>
<td></td>
</tr>
<tr>
<td>MatchMode: MatchMode</td>
<td>• MatchMode—Match mode, including matched by localhash, matched by globalhash, and matched by either localhash or globalhash.</td>
</tr>
<tr>
<td>AllowConnection: number</td>
<td>• AllowConnection—Allowed number of sub sessions.</td>
</tr>
<tr>
<td>ProtocolType: ProtoType</td>
<td>• ProtocolType—Protocol type.</td>
</tr>
<tr>
<td>ApplicationType: type</td>
<td>• ApplicationType—Application-layer protocol type.</td>
</tr>
<tr>
<td>ProtocolType: ProtoType</td>
<td></td>
</tr>
<tr>
<td>ApplicationType: type</td>
<td></td>
</tr>
</tbody>
</table>
L4 LB session backed up. VPN ID: VPN-ID, Virtual Service (IP: IP-address, Port: Port), SNat (IP: IP-address, Port: Port), Real Service (IP: IP-address, Port: Port, Name: rservicename), Client (IP: IP-address, Port: Port)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB backup information. L4 LB session backup succeeded.</td>
<td></td>
</tr>
<tr>
<td>• VPN-ID—VPN ID.</td>
<td></td>
</tr>
<tr>
<td>• Virtual Service—Virtual service information:</td>
<td></td>
</tr>
<tr>
<td>o IP-address—IP address of the virtual service.</td>
<td></td>
</tr>
<tr>
<td>o Port—Port number of the virtual service.</td>
<td></td>
</tr>
<tr>
<td>• SNat—Source address translation information:</td>
<td></td>
</tr>
<tr>
<td>o IP-address—IP address after translation.</td>
<td></td>
</tr>
<tr>
<td>o Port—Port number after translation.</td>
<td></td>
</tr>
<tr>
<td>• Real Service—Real service information:</td>
<td></td>
</tr>
<tr>
<td>o IP-address—IP address of the real service.</td>
<td></td>
</tr>
<tr>
<td>o Port—Port number of the real service.</td>
<td></td>
</tr>
<tr>
<td>o rservicename—Real service name.</td>
<td></td>
</tr>
<tr>
<td>• Client—Client information:</td>
<td></td>
</tr>
<tr>
<td>o IP-address—IP address of the client.</td>
<td></td>
</tr>
<tr>
<td>o Port—Port number of the client.</td>
<td></td>
</tr>
</tbody>
</table>

L4 LB session restored. VPN ID: VPN-ID, Virtual Service (IP: IP-address, Port: Port), SNat (IP: IP-address, Port: Port), Real Service (IP: IP-address, Port: Port, Name: rservicename), Client (IP: IP-address, Port: Port)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB restoration information. L4 LB session was restored:</td>
<td></td>
</tr>
<tr>
<td>• VPN-ID—VPN ID.</td>
<td></td>
</tr>
<tr>
<td>• Virtual Service—Virtual service information:</td>
<td></td>
</tr>
<tr>
<td>o IP-address—IP address of the virtual service.</td>
<td></td>
</tr>
<tr>
<td>o Port—Port number of the virtual service.</td>
<td></td>
</tr>
<tr>
<td>• SNat—Source address translation information:</td>
<td></td>
</tr>
<tr>
<td>o IP-address—IP address after translation.</td>
<td></td>
</tr>
<tr>
<td>o Port—Port number after translation.</td>
<td></td>
</tr>
<tr>
<td>• Real Service—Real service information:</td>
<td></td>
</tr>
<tr>
<td>o IP-address—IP address of the real service.</td>
<td></td>
</tr>
<tr>
<td>o Port—Port number of the real service.</td>
<td></td>
</tr>
<tr>
<td>o rservicename—Real service name.</td>
<td></td>
</tr>
<tr>
<td>• Client—Client information:</td>
<td></td>
</tr>
<tr>
<td>o IP-address—IP address of the client.</td>
<td></td>
</tr>
<tr>
<td>o Port—Port number of the client.</td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB backup information. L7 LB session backup succeeded:</td>
<td></td>
</tr>
<tr>
<td>• Redirect-info—Redirection identifier.</td>
<td></td>
</tr>
<tr>
<td>• Seqnum-adjust—Sequence number adjustment identifier.</td>
<td></td>
</tr>
<tr>
<td>• MSS-info—MSS information.</td>
<td></td>
</tr>
<tr>
<td>• Request Sequence Info—Requested sequence number information:</td>
<td></td>
</tr>
<tr>
<td>o init-info—Initial information.</td>
<td></td>
</tr>
<tr>
<td>o adjust-len—Adjusted length.</td>
<td></td>
</tr>
<tr>
<td>o nextadjust-len—Next adjusted length.</td>
<td></td>
</tr>
<tr>
<td>• Response Sequence Info—Replied sequence number information.</td>
<td></td>
</tr>
</tbody>
</table>
## Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LB LastHop backed up. LastHopFlag: lastHop-flag, SubLastHopFlag: sublasthop-flag, Interface: interface-name, MAC: MAC-address.</th>
<th>LB backup information. Last hop backup succeeded:</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB LastHop restored. LastHopFlag: lastHop-flag, SubLastHopFlag: sublasthop-flag, Interface: interface-name, MAC: MAC-address.</td>
<td>LB restoration information. Last hop restored:</td>
</tr>
</tbody>
</table>

| LB persist backed up. Virtual service ID: VS-ID, Persistence Value: persist-value. | LB persistent backup succeeded: |

### Examples

#### # Enable stateful failover service event debugging.

```bash
<Sysname> debugging dhbk service event
*July 26 09:40:56:763 2007 Sysname DPDHBKS/7/debug:
dhk service event:
  Respond: start sending or receiving backup info.

// Stateful failover started sending or receiving backup data.
```

#### # Enable stateful failover service object information debugging. Output similar to the following example is generated when the client at 10.1.1.11 accesses the FTP server at 100.0.0.100 through FW1 under the following conditions:

- The same one-to-one NAT entry is configured on FW1 and FW2.
- The internal IP address 10.1.1.11 is mapped to the public IP address 100.0.0.50.
- FW1 and FW2 are in **Synchronize** state.

```bash
<Sysname> debugging dhbk service info
*Apr 26 12:46:20:463 2007 Sysname SESS_DP/7/TABLE:
dhk service info:
  Session information:
  ReqTuple5: 10.1.1.11/2798-->100.0.0.100/21(TCP)
  RespTuple5: 100.0.0.100/21-->100.0.0.50/2798(TCP)
```
// A session between 10.1.1.11 and 100.0.0.100 was established. TCP is used for communication.

*Apr 26 12:46:20:464 2000 Sysname DPNAT/7/debug:
  dhbk service info:
  NAT attached information:
  Local IP/Port: 10.1.1.11/0
  Global IP/Port: 100.0.0.50/0
  PatType: STATIC
  VPN: 0

// The IP address and port number 10.1.1.11 and 0 were translated into 100.0.0.50 and 0, respectively by using static NAT.

*Apr 26 12:46:20:464 2000 Sysname DPALG/7/debug:
  dhbk service info:
  ALG attached information:
  FSM: FTP Initition

// The attached information was backed up as the appended data of the session backup data.

# Enable stateful failover service FSM debugging, and then use a command to remove a session.
<Sysname> debugging dhbk service fsm

*July 26 09:55:56:763 2007 Sysname SESS_DP/7/TABLE:
  dhbk service fsm:
  received compulsive aging command
  age session
  sent compulsive aging message

// The aging command was received and the session was then aged out. Then, a compulsive aging message was sent to the peer end.
FSM: ACTIVE --> NULL

// The session entered the NULL queue from the ACTIVE queue.
ReqTuple6:192.168.0.2/3840-->192.168.1.58/3840(TCP)VPN:1
RespTuple6:192.168.1.58/3840-->192.168.0.2/3840(TCP)VPN:2

// The session between 192.168.0.2/3840 and 192.168.1.58/3840 was removed. The VPN numbers of the initiator and receiver are 1 and 2, respectively. TCP is used for communication.

# Enable stateful failover service error debugging to determine that the NAT attached information failed to be backed up.
<Sysname> debugging dhbk service error

*July 26 19:55:56:763 2007 Sysname DPNAT/7/debug:
  dhbk service error:
  Backup NAT attached info failed

// DHBK failed to back up NAT attached information.
Terminal access debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging rta error

Use **debugging rta error** to enable terminal access error debugging.

Use **undo debugging rta error** to disable terminal access error debugging.

**Syntax**

```
debugging rta error
undo debugging rta error
```

**Default**

Terminal access error debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Usage guidelines**

Table 1 describes output fields and messages for the **debugging rta error** command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Telnet Do APP Message: Terminal[n] APP[m] receive AsyncMsgEvtType[u], set APP kept | An error occurred when a Telnet virtual terminal processed an ARP message. The ARP m of terminal n receives asynchronous messages u, and the ARP state is set to kept. The ARP state can be one of the following:  
  - **Kept**—No connection is established.  
  - **Linking**—A connection is being established.  
  - **Linked**—A connection has been established. |

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| SSH Do APP Message: Terminal[n] APP[m] receive RTA_SSHMsgEvtType[u], set APP kept | An error occurred when an SSH virtual terminal processed an ARP message. The ARP m of terminal n receives asynchronous messages u, and the ARP state is set to kept. The ARP state can be one of the following:  
  - **Kept**—No connection is established.  
  - **Linking**—A connection is being established.  
  - **Linked**—A connection has been established. |
Examples

# Enable terminal monitoring and terminal access error debugging. Output similar to the following example is generated when you enter VTY 0 view under the following conditions:

- VTY 0 of telnet type is configured in terminal template view.
- A non-existent IP address is configured for the peer end.
- The template is applied to an interface and the terminal number is set to 1.

```
<Sysname> terminal monitor
<Sysname> terminal debugging
<Sysname> debugging rta error

*Nov 25 17:25:38:546 2006 Sysname RTA/7/debug:
  17:25:38  Create APP Link: Socket connect error!

// Terminal access failed to create the ARP link because of a socket connection error.
*Nov 25 17:25:38:556 2006 Sysname RTA/7/debug:

// An error occurred when telnet processed the ARP message. ARP 2 of terminal 1 received an asynchronous message of type 4, and the ARP state was set to kept. ARP 2 corresponds to VTY 0.
```

debugging rta event

Use `debugging rta event` to enable terminal access event debugging.

Use `undo debugging rta event` to disable terminal access event debugging.

**Syntax**

```
debugging rta event
undo debugging rta event
```

**Default**

Terminal access debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Usage guidelines**

Table 2 describes output fields and messages for the `debugging rta event` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal Receive data flag is NOTRECVDATA!</td>
<td>The data flag received by the terminal is NOTRECVDATA, which means the data cannot be received.</td>
</tr>
</tbody>
</table>
Terminal[n] set state to MENU!

APP[m] initialize ETelnet taskdata successfully.

Terminal[n]: APP[m] free ETelnet taskdata successfully.

Examples

# # Enable terminal monitoring and terminal access event debugging. Output similar to the following example is generated when you enter VTY 0 view and wait for five seconds under the following conditions:

• VTY 0 of telnet type is configured in terminal template view.
• The auto-link timer is set to 5 seconds.
• The template is applied to an interface and the terminal number is set to 1.

<Sysname> terminal monitor
<Sysname> terminal debugging
<Sysname> debugging rta event
<Sysname>

  APP[4] link was created upon expiration of auto-link timer of terminal[1]!
  // The link of ARP 4 was created upon expiration of auto-link timer of terminal 1.
*Nov 25 17:46:24:735 2006 Sysname RTA/7/debug:
  APP[4] create TCP socket successfully!
  // The TCP socket was created by ARP 4.

debugging rta packet

Use debugging rta packet to enable terminal access packet debugging.

Use undo debugging rta packet to disable terminal access packet debugging.

Syntax

debugging rta packet { brief | detail } { all | recv-remote | recv-terminal | send-remote | send-terminal } terminal-number

undo debugging rta packet { brief | detail } { all | recv-remote | recv-terminal | send-remote | send-terminal } terminal-number

Default

Terminal access packet debugging is disabled.
Views

User view

Default command level

1: Monitor level

Parameters

- **brief**: Specifies brief terminal access packet information debugging.
- **detail**: Specifies detailed terminal access packet information debugging.
- **all**: Specifies all types of terminal access packet debugging.
- **recv-remote**: Displays information about terminal access packets received from the peer end.
- **recv-terminal**: Displays information about terminal access packets received from the terminal.
- **send-remote**: Displays information about terminal access packets sent to the peer end.
- **send-terminal**: Displays information about terminal access packets sent to the terminal.
- **terminal-number**: Specifies a terminal by its number.

Examples

# Enable terminal monitoring and terminal access brief packet information debugging. Output similar to
# the following example is generated when you perform operations on the terminal under the following
# conditions:

- The terminal number is set to 1.
- A link to a FEP is established.

```bash
<Sysname> terminal monitor
<Sysname> terminal debugging
<Sysname> debugging rta packet brief all 1
<Sysname>
*Nov 25 18:02:30:885 2006 Sysname RTA/7/debug:
  18:02:30  Terminal[1]: Receive from terminal 1 bytes.
*Nov 25 18:02:30:886 2006 Sysname RTA/7/debug:
  18:02:30  Terminal[1]: Send to socket 1 bytes.
*Nov 25 18:02:30:895 2006 Sysname RTA/7/debug:
  18:02:30  Terminal[1]: Receive from remote 2 bytes.
*Nov 25 18:02:30:896 2006 Sysname RTA/7/debug:
  18:02:30  Terminal[1]: Send to terminal 2 bytes.
*Nov 25 18:02:30:916 2006 Sysname RTA/7/debug:
  18:02:30  Terminal[1]: Send to terminal 2 bytes.
```

// Terminal access received one byte of data from the terminal and sent it to the FEP. Terminal access
received two bytes of data from the peer end and sent it the terminal. Terminal access received 19 bytes
of data from the FEP and sent it to the terminal.

# Enable terminal monitoring and terminal access detailed packet information debugging. Output
similar to the following example is generated when you perform operations on the terminal under the
following conditions:

- The terminal number is set to 1.
• A link to a FEP is established.

<Sysname> terminal monitor
<Sysname> terminal debugging
<Sysname> debugging rta packet detail recv-terminal 1
<Sysname>
*Nov 25 19:25:57:712 2006 Sysname RTA/7/debug:
  19:25:57  Terminal[1]: Receive from terminal 1 bytes:
*Nov 25 19:25:57:713 2006 Sysname RTA/7/debug:
  -----------------------------------------------
  000: 0d
  // Terminal access received one byte of data (0d) from the terminal.

<Sysname> debugging rta packet detail recv-remote 1
<Sysname>
*Nov 25 19:28:44:633 2006 Sysname RTA/7/debug:
  19:28:44  Terminal[1]: Receive from remote 2 bytes:
*Nov 25 19:28:44:633 2006 Sysname RTA/7/debug:
  -----------------------------------------------
  000: 0d 0a
  // Terminal access received two bytes of data (0d 0a) from the FEP.
Track debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging track

Use debugging track to enable Track debugging.

Use undo debugging track to disable Track debugging.

Syntax

debugging track

undo debugging track

Default

Track debugging is disabled.

Views

User view

Default command level

1: Monitor level

Usage guidelines

Table 1 describes the output fields and messages for the debugging track command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify application module(module-id) that the status of track entry entry-number has changed from state1 to state2.</td>
<td>Track entry state:</td>
</tr>
<tr>
<td></td>
<td>1—Invalid.</td>
</tr>
<tr>
<td></td>
<td>2—Positive, indicating normal state.</td>
</tr>
<tr>
<td></td>
<td>3—Negative, indicating abnormal state.</td>
</tr>
</tbody>
</table>
### Field

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;timestamp&quot; indicates the time when the message was sent, in the format of <code>hh:mm:ss:sss</code>, for example, <code>10:12:21:708</code>.</td>
</tr>
<tr>
<td><code>hh</code> is in the range of 00 to 23, <code>mm</code> and <code>ss</code> are in the range of 00 to 59, and <code>sss</code> is in the range of 0 to 999.</td>
</tr>
<tr>
<td><code>track-status</code> values:</td>
</tr>
<tr>
<td>Positive.</td>
</tr>
<tr>
<td>Invalid.</td>
</tr>
<tr>
<td>Negative.</td>
</tr>
<tr>
<td><code>message-type</code> values:</td>
</tr>
<tr>
<td>1 — Creates a track object.</td>
</tr>
<tr>
<td>2 — Deletes a track object.</td>
</tr>
<tr>
<td>3 — Changes the status of a track object.</td>
</tr>
</tbody>
</table>

#### Examples

1. **Enable debugging for the Track module.**
   ```
   <Sysname> terminal debugging
   <Sysname> terminal monitor
   <Sysname> debugging track
   ```
2. **Create an NQA test group, and configure the test type, test parameters, and the reaction entry.**
   ```
   <Sysname> system-view
   [Sysname] nqa entry admin test
   [Sysname-nqa-admin-test] type icmp-echo
   [Sysname-nqa-admin-test-icmp-echo] destination ip 2.2.2.2
   ```
[Sysname-nqa-admin-test-icmp-echo] frequency 5000
[Sysname-nqa-admin-test-icmp-echo] reaction 1 checked-element probe-fail threshold-type consecutive 5 action-type trigger-only

*Apr 30 18:58:56:520 2007 Sysname TRACK/7/TRACK Debug: Receive the notification that NQA(admin-test) reaction(1) is created.

// Track received a notification that NQA (admin-test) Reaction entry 1 had been created.

# Create a track entry.

[Sysname-nqa-admin-test-icmp-echo] quit
[Sysname] track 1 nqa entry admin test reaction 1

*Apr 30 19:09:20:2007 Sysname TRACK/7/TRACK Debug: Notify application module(0x4240000) that track entry 1 is created.

// Track notified the backup center that track object 1 had been created.

*Apr 30 19:09:20:500 2007 Sysname TRACK/7/TRACK Debug: Notify application module(0x5230000) that track entry 1 is created.

// Track notified the VRRP module that track object 1 had been created.

*Apr 30 19:09:20:515 2007 Sysname TRACK/7/TRACK Debug: Notify application module(0x4470000) that track entry 1 is created.

// Track notified the policy routing module that track object 1 had been created.

*Apr 30 19:09:20:515 2007 Sysname TRACK/7/TRACK Debug: Notify application module(0x6020000) that track entry 1 is created.

// Track notified the static routing module that track object 1 had been created.

# Configure a static route, and verify the reachability of the next hop of the static route.

[Sysname] ip route-static 1.1.1.1 16 2.2.2.2 track 1

# Schedule the NQA test group.

[Sysname] nqa schedule admin test start-time now lifetime forever

*Apr 30 19:12:31:738 2007 Sysname TRACK/7/TRACK Debug: Receive the notification that the status of NQA(admin-test) reaction(1) has changed to 2.

// NQA notified the Track module of the state change of the NQA (admin-test) Reaction entry 1 to 2, indicating that the next hop of the static route is unreachable.

*Apr 30 19:12:31:738 2007 Sysname TRACK/7/TRACK Debug: Notify application module(0x6020000) that the status of track entry 1 has changed from 1 to 3.

// Track notified the static routing module of the state change of track entry 1 from 1 to 3, indicating that the next hop of the static route is unreachable.
Tunneling debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging ipv4-tunnel

Use `debugging ipv4-tunnel` to enable IPv4 tunnel debugging.

Use `undo debugging ipv4-tunnel` to disable IPv4 tunnel debugging.

**Syntax**

```
debugging ipv4-tunnel { all | error | packet }
undo debugging ipv4-tunnel { all | error | packet }
```

**Default**

No IPv4 tunnel debugging is enabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **all**: Specifies all IPv4 tunnel debugging.
- **error**: Specifies IPv4 tunnel error debugging.
- **packet**: Specifies IPv4 tunnel packet debugging.

**Usage guidelines**

Table 1 describes the output fields and messages for the `debugging ipv4-tunnel error` command (on single-core devices).

**Table 230 Output from the debugging ipv4-tunnel error command (on single-core devices)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnel num error: Failed to transmit packet</td>
<td>IPv4 tunneling failed to send a tunnel packet from tunnel interface num.</td>
</tr>
<tr>
<td>Invalid IPv4 source address. Dropped the packet</td>
<td>The outbound packet was discarded because the source address in the IPv4 header is invalid.</td>
</tr>
<tr>
<td>Invalid IPv6 source address. Dropped the packet</td>
<td>The outbound packet was discarded because the source address in the IPv6 header is invalid.</td>
</tr>
<tr>
<td>No tunnel in the state of UP was found for the packet. Dropped the packet</td>
<td>The packet was discarded because no tunnel in the up state is found for decapsulating the packet.</td>
</tr>
</tbody>
</table>

Table 2 describes the output fields and messages for the `debugging ipv4-tunnel error` command (on multi-core devices).
### Table 231 Output from the debugging ipv4-tunnel error command (on multi-core devices)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnel interface is down. Dropped the packet</td>
<td>The packet was discarded because the tunnel is down.</td>
</tr>
</tbody>
</table>

### Table 3 describes the output fields and messages for the debugging ipv4-tunnel packet command (on single-core devices).

### Table 232 Output from the debugging ipv4-tunnel packet command (on single-core devices)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnelm packet:Before encapsulation, Outer packet header ipaddress1 -&gt; ipaddress2(length = length)</td>
<td>The source address and destination address of the packet to be encapsulated on interface Tunnelm are ipaddress1 and ipaddress2, respectively. The length of the packet is length.</td>
</tr>
<tr>
<td>Tunnelm packet:After encapsulation, Outer packet header ipaddress1 -&gt; ipaddress2(length = length)</td>
<td>The source address and destination address of the packet after encapsulation on interface Tunnelm are ipaddress1 and ipaddress2, respectively. The length of the packet is length.</td>
</tr>
<tr>
<td>Ipv4-tunnel_packet: Decapsulate tunnel packet Incoming packet header ipaddress1 -&gt; ipaddress2(length = length)</td>
<td>The source address and destination address of the packet to be decapsulated are ipaddress1 and ipaddress2, respectively. The length of the packet is length.</td>
</tr>
<tr>
<td>Tunnelm packet:After decapsulation, Outgoing packet header ipaddress1 -&gt; ipaddress2(length = length)</td>
<td>The source address and destination address of the packet after decapsulation on interface Tunnelm are ipaddress1 and ipaddress2, respectively. The length of the packet is length.</td>
</tr>
</tbody>
</table>

### Table 4 describes the output fields and messages for the debugging ipv4-tunnel packet command (on multi-core devices).

### Table 233 Output from the debugging ipv4-tunnel packet command (on multi-core devices)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnelnum packet: Before encapsulation, packet header source-address -&gt; destination-address(length = length)</td>
<td>The source address and destination address of the packet to be encapsulated on interface Tunnelnum are source-address and destination-address, respectively. The length of the packet is length.</td>
</tr>
<tr>
<td>Tunnelnum packet: After encapsulation, outgoing packet header source-address -&gt; destination-address(length = length)</td>
<td>The source address and destination address of the packet after encapsulation on interface Tunnelnum are source-address and destination-address, respectively. The length of the packet is length.</td>
</tr>
<tr>
<td>packet: Before decapsulation, packet header source-address -&gt; destination-address(length = length)</td>
<td>The source address and destination address of the packet to be decapsulated are source-address and destination-address, respectively. The length of the packet is length.</td>
</tr>
<tr>
<td>Tunnelnum packet: After decapsulation, outgoing packet header source-address -&gt; destination-address(length = length)</td>
<td>The source address and destination address of the packet after decapsulation on interface Tunnelnum are source-address and destination-address, respectively. The length of the packet is length.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>packet: Packet is sent to slot number</td>
<td>A tunnel packet was sent to interface board number for decapsulation.</td>
</tr>
</tbody>
</table>

**Examples**

- **On single-core devices:**

  # Enable IPv4 tunnel error debugging on the local device. Output similar to the following example is generated when an IPv4 packet with an invalid source IP address (a loopback interface address, for example) arrives under the following conditions:
  - An IPv4 tunnel is established between two devices.
  - Parameters are configured to bring up the tunnel interfaces.
  - The packet is sent through the tunnel from the local end.

    ```
    <Sysname> debugging ipv4-tunnel error
    *0.253156 Sysname IPV4TUNN/7/debug:
    Tunnel1 error: Invalid IPv4 source address. Dropped the packet.
    // IPv4 tunneling discarded the packet at interface Tunnel 1 due to an invalid source IP address.
    ```

  # Enable IPv4 tunnel packet debugging on the local device. Output similar to the following example is generated when you execute the `ping` command under the following conditions:
  - An IPv4 tunnel is established between two devices.
  - Parameters are configured to bring up the tunnel interfaces.

    ```
    <Sysname> debugging ipv4-tunnel packet
    <Sysname> ping –c 1 5.5.5.2
    PING 5.5.5.2: 56 data bytes, press CTRL_C to break
    *0.2812406 Sysname IPV4-TUN/8/debug:
    Tunnel1 packet:Before encapsulation, 
    Outer packet header 5.5.5.1->5.5.5.2(length = 84)
    // Before encapsulation at interface Tunnel 1, the source and destination IP addresses of the packet header are 5.5.5.1 and 5.5.5.2, and the packet length is 84.
    *0.2812422 Sysname IPV4-TUN/8/debug:
    Tunnel1 packet:After encapsulation, 
    Outer packet header 192.168.19.41->192.168.19.42(length = 104)
    // After encapsulation at interface Tunnel 1, the source and destination IP addresses of the packet header are 192.168.19.41 and 192.168.19.42, and the packet length is 104.
    *0.2812468 Sysname IPV4-TUN/8/debug:
    ipv4-tunnel_packet: Decapsulate tunnel packet
    // Before decapsulation at interface Tunnel 1, the source and destination IP addresses of the packet header are 192.168.19.42 and 192.168.19.41, and the packet length is 104.
    *0.2812468 Sysname IPV4-TUN/8/debug:
    Tunnel1 packet:After decapsulation, 
    Outgoing packet header 5.5.5.2->5.5.5.1(length = 84)
    // After decapsulation at interface Tunnel 1, the source and destination IP addresses of the packet header are 5.5.5.2 and 5.5.5.1, and the packet length is 84.
    Reply from 5.5.5.2: bytes=56 Sequence=1 ttl=255 time=78 ms
    ```
--- 5.5.5.2 ping statistics ---
1 packet(s) transmitted
1 packet(s) received
0.00% packet loss
round-trip min/avg/max = 78/78/78 ms

- On multi-core devices:

  # Enable IPv4 tunnel error debugging on the local device. The output in this example was created when the following conditions exist:
  - An IPv4 tunnel is established between two devices.
  - Parameters are configured to bring up only the peer tunnel interface.

```
<Sysname> debugging ipv4-tunnel error
*Mar  6 11:56:46:312 2008 Sysname DP4IN4/7/debug:
error: Failed to find tunnel interface.
```

  // The local end failed to find the tunnel interface for decapsulation of the packet sent from the peer end.

  # Enable IPv4 tunnel packet debugging on the local device. Output similar to the following example is generated when you execute the ping command under the following conditions:
  - An IPv4 tunnel is established between two devices.
  - Parameters are configured to bring up the tunnel interfaces.

```
<Sysname> debugging ipv4-tunnel packet

<Sysname> ping -c 1 22.1.1.2
PING 22.1.1.2: 56 data bytes, press CTRL_C to break
*Mar  6 11:03:21:578 2008 Sysname DP4IN4/7/debug:
Tunnel0 packet: Before encapsulation,
packet header: 1.1.1.1->22.1.1.2(length = 84).
```

  // Before encapsulation at interface Tunnel 0, the source and destination IP addresses of the packet header are 1.1.1 and 22.1.1.2, and the packet length is 84.

```
*Mar  6 11:03:21:578 2008 Sysname DP4IN4/7/debug:
Tunnel0 packet: After encapsulation,
outgoing packet header 2.1.1.1->2.1.1.2(length = 104).
```

  // After encapsulation at interface Tunnel 0, the source and destination IP addresses of the packet header are 2.1.1.1 and 2.1.1.2, and the packet length is 104.

```
*Mar  6 11:03:21:593 2008 Sysname DP4IN4/7/debug:
packet: Before decapsulation,
packet header 2.1.1.2->2.1.1.1(length = 104).
```

  // Before decapsulation at interface Tunnel 0, the source and destination IP addresses of the packet header are 2.1.1.2 and 2.1.1.1, and the packet length is 104.

```
*Mar  6 11:03:21:593 2008 Sysname DP4IN4/7/debug:
Tunnel0 packet: After decapsulation,
outgoing packet header 22.1.1.2->1.1.1.1(length = 84).
```

  // After decapsulation at interface Tunnel 0, the source and destination IP addresses of the packet header are 22.1.1.2 and 1.1.1.1, and the packet length is 84.
debugging ipv6-tunnel

Use **debugging ipv6-tunnel** to enable IPv6 tunnel debugging.

Use **undo debugging ipv6-tunnel** to disable IPv6 tunnel debugging.

**Syntax**

```
debugging ipv6-tunnel { all | error | packet }
undo debugging ipv6-tunnel { all | error | packet }
```

**Default**

No IPv6 tunnel debugging is enabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **all**: Specifies all IPv6 tunnel debugging.
- **error**: Specifies IPv6 tunnel error debugging.
- **packet**: Specifies IPv6 tunnel packet debugging.

**Usage guidelines**

Table 5 describes the output fields and messages for the **debugging ipv6-tunnel error** command (on single-core devices).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting destination failed</td>
<td>IPv6 tunnel failed to obtain the destination IPv4 address embedded in the inbound IPv6 tunnel packet.</td>
</tr>
<tr>
<td>fragment check</td>
<td>The IPv6 fragment is checked according to RFC 2893.</td>
</tr>
<tr>
<td>IPv4 transmit failed</td>
<td>IPv6 tunnel failed to send the encapsulated IPv4 tunnel packet through the tunnel interface.</td>
</tr>
<tr>
<td>IPv6 transmit failed</td>
<td>IPv6 tunnel failed to send the encapsulated IPv6 tunnel packet through the tunnel interface.</td>
</tr>
<tr>
<td>Adding IPv4 header failed</td>
<td>IPv6 tunnel failed to add the IPv4 header to the IPv6 tunnel packet.</td>
</tr>
<tr>
<td>Adding IPv6 header failed</td>
<td>IPv6 tunnel failed to add the IPv6 header to the IPv6 tunnel packet.</td>
</tr>
<tr>
<td>Fragment checking of the IPv6 packet failed</td>
<td>IPv6 tunnel failed to check the IPv6 fragment according to RFC 2893.</td>
</tr>
</tbody>
</table>

Table 6 describes the output fields and messages for the **debugging ipv6-tunnel error** command (on multi-core devices).
### Table 235 Output from the debugging ipv6-tunnel error command (on multi-core devices)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to add IPv4 header</td>
<td>IPv6 tunnel failed to add the IPv4 header to the IPv6 tunnel packet.</td>
</tr>
<tr>
<td>Failed to add IPv6 header</td>
<td>IPv6 tunnel failed to add the IPv6 header to the IPv6 tunnel packet.</td>
</tr>
<tr>
<td>Failed to send packet to FC module</td>
<td>IPv6 tunnel failed to classify the flow.</td>
</tr>
</tbody>
</table>

### Examples

- On single-core devices:

  ```
  # Establish an IPv6 6to4 tunnel between two devices and configure parameters to bring up the tunnel interfaces. (Details not shown.)
  # Configure a static IPv6 route with interface Tunnel 2 as the outgoing interface.
  <Sysname> system-view
  [Sysname] ipv6 route-static 111::1 16 Tunnel 2
  # Enable IPv6 tunnel error debugging on the local device. The output in this example was created when IPv6 tunneling sent an IPv6 packet to destination 111::/16 through the 6to4 tunnel:
  <Sysname> debugging ipv6-tunnel error
  *0.2245453 Sysname IPV6TUN/7/debug:
  Tunnel1 error: Adding IPv4 header failed.
  // No destination IPv4 address can be obtained from the IPv6 packet. IPv6 tunneling failed to add an IPv4 header to the packet.
  # Enable IPv6 tunnel packet debugging on the local device. Output similar to the following example is generated when you execute the ping command under the following conditions:
  - An IPv6 tunnel is established between two devices.
  - Parameters are configured to bring up the tunnel interfaces.
  <Sysname> debugging ipv6-tunnel packet
  <Sysname> ping ipv6 -c 1 2005::2
  PING 2005::2 : 56 data bytes, press CTRL_C to break
  *0.3760250 Sysname IPV6-TUN/8/debug:
  Tunnel2 packet: Before encapsulation,
  Incoming packet header 2005::0001->2005::0002(length = 104)
  // Before encapsulation at the tunnel interface Tunnel 2, the source and destination IP addresses in the packet header are 2005::0001 and 2005::0002, and the packet length is 104.
  *0.3760265 Sysname IPV6-TUN/8/debug:
  Tunnel2 packet: After encapsulation,
  Outgoing packet header 2003::0001->2003::0002(length = 144)
  // After encapsulation at the tunnel interface Tunnel 2, the source and destination IP addresses in the packet header are 2003::0001 and 2003::0002, and the packet length is 144.
  *0.3760297 Sysname IPV6-TUN/8/debug:
  ipv6-tunnel_event:transproto is ipv6.
  *0.3760297 Sysname IPV6-TUN/8/debug:
  ipv6-tunnel_packet: Decapsulate tunnel packet
  Incoming packet header 2003::0002->2003::0001(length = 144)
  ```
Before decapsulation at the tunnel interface Tunnel 2, the source and destination IP addresses in the packet header are 2003::0002 and 2003::0001, and the packet length is 144.

*0.3760312 Sysname IPV6-TUN/8/debug:
  Tunnel2 packet: After decapsulation,
    Outgoing packet header 2005::0002->2005::0001(length = 104)

After encapsulation at the tunnel interface Tunnel 2, the source and destination IP addresses in the packet header are 2005::0002 and 2005::0001, and the packet length is 104.

Reply from 2005::2: Bytes=56 Sequence=1 hop limit=255  time = 78 ms

--- 2005::2 ping statistics ---
  1 packet(s) transmitted
  1 packet(s) received
  0.00% packet loss
round-trip min/avg/max = 78/78/78 ms

On multi-core devices:

# Establish an IPv6 6to4 tunnel between two devices and configure parameters to bring up the tunnel interfaces. (Details not shown.)

# Configure a static IPv6 route with interface Tunnel 1 as the outgoing interface.

<Sysname> system-view
[Sysname] ipv6 route-static 2000:: 16 tunnel 1

# Enable IPv6 tunnel error debugging on the local device. The output in this example was created when an IPv6 packet to destination 2000:1234::1 is sent through the 6to4 tunnel:

<Sysname> debugging ipv6-tunnel error
*Mar 6 13:02:40:734 2008 Sysname DP6IN4/7/debug:
  error: IPv6 destination address is not 6to4 address.
*Mar 6 13:02:40:750 2008 Sysname DP6IN4/7/debug:
  Tunnel1 error: Failed to get the tunnel's destination address.

No destination IPv4 address can be obtained from the IPv6 packet due to an incorrect 6to4 address format.

# Modify the static route configuration.
[Sysname] ipv6 route-static 2000:: 16 2002:201:102::1
[Sysname] quit

# Enable IPv6 tunnel packet debugging. The output in this example was created when you execute the ping command:

<Sysname> debugging ipv6-tunnel packet
<Sysname> ping ipv6 -c 1 2000:1234::1
PING 2000:1234::1 56 data bytes, press CTRL_C to break
*Mar 6 13:08:34:875 2008 Sysname DP6IN4/7/debug:
  Tunnel1 packet: Before encapsulation,

// Before encapsulation at the tunnel interface Tunnel 1, the source and destination IP addresses in the packet header are 1:1 and 2000:1234::1, and the packet length is 104.

*Mar 6 13:08:34:890 2008 Sysname DP6IN4/7/debug:
  Tunnel1 packet: After encapsulation,
    outgoing packet header 2.1.1.1->2.1.1.2(length = 124).
debugging tunnel

Use `debugging tunnel` to enable tunnel debugging.

Use `undo debugging tunnel` to disable tunnel debugging.

Syntax

```
debugging tunnel { all | error | event | packet }
undo debugging tunnel { all | error | event | packet }
```

Default

Tunnel debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

- `all`: Specifies all tunnel debugging.
- `error`: Specifies Tunnel error debugging.
- `event`: Specifies Tunnel event debugging.
- `packet`: Specifies Tunnel packet debugging.

Examples

```
# Enable tunnel error debugging on the local device. The output in this example was created when the following conditions exist:
  • Tunnel interfaces have been created on two devices.
  • The tunnel interface is in up state, but failed to notify the driver.
<Sysname> debugging tunnel error
*Jul 17 09:16:07:928 2008 Sysname TUNNEL/7/debug:
  Failed to notify the driver of the interface up event
// The tunnel interface failed to notify the drive of the event that the interface is up.
# Create tunnel interfaces on two devices, and enable tunnel event debugging on the local device.
```
<Sysname> debugging tunnel event
*Jul 17 09:16:07:930 2008 Sysname TUNNEL/7/debug:
   Tunnel1 can't come up because:
   // The link state of Tunnel1 cannot be up, and the reason might be the following:
*Jul 17 09:16:07:930 2008 Sysname TUNNEL/7/debug:
   Tunnel source is null.
   // The link state of Tunnel1 was down because the source IP address was not configured.
*Jul 17 09:16:07:935 2008 Sysname TUNNEL/7/debug:
   Tunnel destination is null.
   // The link state of Tunnel1 was down because the destination address was not configured.

# Enable tunnel packet debugging on the local device. The output in this example was created when the
following conditions exist:
   • Tunnel interfaces have been created on two devices.
   • For some switches, packet decapsulation is implemented through chips. After being received by the
     local device, the tunnel packets are decapsulated, and then sent to the tunnel interface for
     processing.

<Sysname> debugging tunnel packet
*Jul 17 09:16:07:928 2008 Sysname TUNNEL/7/debug:
   Tunnel receive packet(bDecapsulate d = 1), tunnel index = 0x32f0000.
   // The tunnel interface received a decapsulated packet delivered by the drive.
UDP helper debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

**debugging udp-helper**

Use `debugging udp-helper` to enable UDP helper debugging.

Use `undo debugging udp-helper` to disable UDP helper debugging.

**Syntax**

```plaintext
debugging udp-helper { event | packet [ receive | send ] }
undo debugging udp-helper { event | packet [ receive | send ] }
```

**Default**

UDP helper debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- `event`: Specifies UDP helper event debugging.
- `packet`: Specifies UDP helper packet debugging.
- `receive`: Specifies UDP helper debugging for received packets.
- `send`: Specifies UDP helper debugging for sent packets.

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging udp-helper event` command.

**Table 236 Output from the debugging udp-helper event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBUF copy failed</td>
<td>UDP helper failed to copy packets.</td>
</tr>
<tr>
<td>Prompt: TTL Exceed!</td>
<td>TTL expired.</td>
</tr>
<tr>
<td>Dest Ip(Num) is not routable or the packet is from the device itself.</td>
<td>No operation was performed because the outgoing interface did not exist or the packet was sent from the local device.</td>
</tr>
<tr>
<td>Pointer of the route table outer interface is NULL!</td>
<td>Outgoing interface pointer of the route entry is NULL.</td>
</tr>
<tr>
<td>Getting self Unit ID failed</td>
<td>UDP helper failed to get the ID of the local unit.</td>
</tr>
<tr>
<td>Prompt: Convert xid to saved xid</td>
<td>UDP helper converted XID and saved it.</td>
</tr>
</tbody>
</table>
Field | Description
---|---
Prompt: Send the packet to udp module in BOOTP and DHCP case | BOOTP or DHCP UDP packets were directly delivered to the UDP module.

Self UNIT is MASTER UNIT
Send bootp/dhcp RESPONSE packet to udp module | The local unit was the active MPU, and BOOTP/DHCP responses were delivered to the UDP module.

Prompt: Self UNIT is SLAVE UNIT
Send bootp/dhcp RESPONSE packet to MASTER UNIT: number | The local unit was a standby MPU, and BOOTP/DHCP responses were delivered to the active MPU. Number specifies the card number of the active MPU.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Dest Ip</td>
<td>New destination IP address of the packet after processed by UDP helper.</td>
</tr>
</tbody>
</table>

Prompt: Intercept a UDP packet. UDP helper received a UDP packet.

New Dest Ip ip-address | UDP helper forwarded the packet to the UDP server whose IP address is ip-address.

Prompt: Forward the packet to destination server |

### Examples

# Enable UDP helper packet debugging.
```bash
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging udp-helper packet
```

# Enable UDP helper.
```bash
<Sysname> system-view
[Sysname] udp-helper enable
```

# Configure UDP helper to forward broadcast packets with the UDP destination port 520 to the destination server 2.2.2.2.
```bash
[Sysname] udp-helper port 520
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] udp-helper server 2.2.2.2
```

Output similar to the following example is generated when a UDP packet with the destination port 520 arrives at Ethernet 1/1:
```bash
*Aug 11 08:20:46:00 2005 Sysname UDPH/7/UDPHelper_Pkt:
Dest Ip(1.1.1.255) Source Ip(1.1.1.5) Dest Port(520)
  Prompt: Intercept a UDP packet.
  // UDP helper received a UDP packet.
*Aug 11 08:20:46:01 2005 Sysname UDPH/7/UDPHelper_Pkt:
Dest Ip(1.1.1.255) Source Ip(1.1.1.5) Dest Port(520)
  Prompt: Copy the packet and send it to the UDP module.
  // UDP helper copied the UDP packet and sent it to the UDP module.
*Aug 11 08:20:46:02 2005 Sysname UDPH/7/UDPHelper_Pkt:
```

460
Dest Ip(1.1.1.255)    Source Ip(1.1.1.5)    Dest Port(520)
New Dest Ip(2.2.2.2)

Prompt: Forward the packet to destination server

// UDP helper sent the UDP packet to the server at 2.2.2.2.
URPF debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging ip urpf discards

Use `debugging ip urpf discards` to enable URPF debugging.

Use `undo debugging ip urpf discards` to disable URPF debugging.

Syntax

```
debugging ip urpf discards [ interface interface-type interface-number ]
undo debugging ip urpf discards [ interface interface-type interface-number ]
```

Default

URPF debugging is disabled.

Views

User view

Default command level

1: Monitor level

Parameters

`interface interface-type interface-number`: Specifies an interface by its type and number.

Usage guidelines

Table 1 describes output fields and messages for the `debugging ip urpf discards` command.

### Table 238 Output from the debugging ip urpf discards command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URPF</td>
<td>URPF-Discard: Packet from ip-address via interface-type interface-number</td>
</tr>
<tr>
<td>URPF</td>
<td>URPF-Discard-Suppress: Packet from ip-address via interface-type interface-number</td>
</tr>
</tbody>
</table>

Examples

# Enable URPF debugging. When a packet with an unknown source IP address arrives at Ethernet 1/1, output similar to the following example is generated:

```
<Sysname> debugging ip urpf discards
<Sysname> terminal monitor
<Sysname> terminal debugging
*0.3933516 Sysname URPF/7/Discard:
  URPF  URPF-Discard: Packet from 2.2.2.5 via Ethernet1/1

// URPF dropped a packet received from Ethernet 1/1. The packet has a source IP address of 2.2.2.5.
```
VLAN termination debugging commands

The VLAN termination module name is identified as "SIFVLAN" in debugging messages. The output description tables in this document only contain fields and messages that require an explanation.

debugging vlan

Use debugging vlan to enable VLAN termination debugging. Use undo debugging vlan to disable VLAN termination debugging.

Syntax

```
debugging vlan { all | error | event | packet } [ interface interface-type interface-number ]
undo debugging vlan { all | error | event | packet } [ interface interface-type interface-number ]
```

Default

All types of VLAN termination debugging are disabled.

Views

User view

Default command level

1. Monitor level

Parameters

- **all**: All debugging functions for VLAN termination.
- **error**: Debugging for errors occurred in VLAN termination.
- **event**: Event debugging for VLAN termination.
- **packet**: Packet debugging for VLAN termination.
- **interface interface-type interface-number**: Specifies an interface by its type and number.

Usage guidelines

Table 1 describes output fields and messages for the debugging vlan command.
Table 239 Output from the debugging vlan command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface-name:</td>
<td>SIFVLAN created subinterface interface-name by calling the create function on the drive. The subinterface state changed from state1 to state2. The VID argument indicates the VID of the interface.</td>
</tr>
</tbody>
</table>
| state1 to state2, DRV create interface, the Vid is VID | • interface-name specifies the subinterface name, for example, Ethernet 1/1.0.  
• state1 and state2 refer to the subinterface states.  
• VID refers to the port VLAN ID of the current subinterface. |
| interface-name:           | Subinterface interface-name was removed and the subinterface state changed from state1 to state2. The port VLAN ID of the subinterface is VID. |
| state1 to state2, DRV destroy interface, the Vid is VID |  |
| interface-name:           | The state of subinterface interface-name changed from state1 to state2. QinQ is enabled on the subinterface. |
| state1 to state2, enable qinq |  |
| interface-name:           | SIFVLAN specified a VLAN ID range for subinterface interface-name by calling the range function on the drive. The number of the nodes contained in the chain is specified by the NUM argument. |
| Setting range by DII, node number is NUM |  |
| interface-name:           | Subinterface interface-name sent out a packet whose length is shown by the length argument and contents by the context argument. |
| OUT packet,len length context | Subinterface interface-name received a packet whose length is specified by the length argument and contents are specified by the context argument. |
| interface-name:           |  |
| IN packet,len length context |  |

Examples

# Enable VLAN termination debugging. Output similar to the following example is created when the following conditions exist:
• Subinterface Ethernet 1/1.1 is created.
• VLAN termination is configured on subinterface Ethernet 1/1.0.
<Sysname> debugging vlan all  
<Sysname> system-view  
[Sysname] interface ethernet 1/1.1  
[Sysname-Ethernet1/1.1] vlan-type dot1q vid 1  
*Feb 24 10:34:57:804 2023 Sysname SIFVLAN/7/EVENT:  
 GigabitEthernet1/1.1:  
None to unique dot1q, DRV create interface, the Vid is 0x1  
// SIFVLAN set the VLAN ID of the subinterface to 1 and created the subinterface from the drive.  
[Sysname-Ethernet1/1.1] undo vlan-type dot1q vid 1  
*Feb 24 10:50:19:644 2023 Sysname SIFVLAN/7/EVENT:  
Ethernet1/1.1:  
Unique dot1q to none, DRV destroy interface, the Vid is 0x1
// SIFVLAN deleted the VLAN ID information configured for the subinterface and removed the subinterface by calling the destroy function on the drive.

```
[Sysname=Ethernet1/1.1] vlan-type dot1q vid 1 second-dot1q 10 12
```

*Mar 26 17:07:25:156 2008 Sysname SIFVLAN/7/EVENT:

```
Ethernet1/1.1:
None to ambiguous qinq, DRV create interface, the Vid is 0x1
```

*Mar 26 17:07:25:156 2008 Sysname SIFVLAN/7/EVENT:

```
Ethernet1/1.1:
None to ambiguous qinq, enable qinq
```

*Mar 26 17:07:25:156 2008 Sysname SIFVLAN/7/EVENT:

```
Ethernet1/1.1:
Setting range by Dll, node number is 2
```

// SIFVLAN configured ambiguous QinQ on the subinterface as follows:

- *Created the subinterface.*
- *Enabled QinQ on the subinterface.*
- *Set the range of the VLAN IDs in the QinQ packets that can be terminated by the subinterface.*

The create, qinq enable, and range functions were called on the drive.

```
[Sysname=Ethernet1/1.1] vlan-type dot1q vid 1 second-dot1q 13
[Sysname=Ethernet1/1.1] ip address 12.1.1.2 255.255.255.0
[Sysname=Ethernet1/1.1] ping -c 1 12.1.1.1
```

*Mar 26 17:27:52:609 2008 Sysname SIFVLAN/7/PACKET:

```
Ethernet1/1.1:
OUT packet, len 50
   ff ff ff ff ff ff 00 e0 14 03 32 00 81 00 00 01
   81 00 00 0d 08 06 00 01 08 00 06 04 00 01 00 e0
   14 03 32 00 0c 01 01 02 00 00 00 00 00 00 0c 01
       01 01
```

// SIFVLAN sent a 50-byte broadcast packet out of subinterface Ethernet 1/1.1.

*Mar 26 17:27:52:671 2008 Sysname SIFVLAN/7/PACKET:

```
Ethernet1/1.1:
IN packet, len 50
   00 e0 14 03 32 00 00 e0 14 03 28 00 81 00 00 01
   81 00 00 0d 08 06 00 01 08 00 06 04 00 02 00 e0
   14 03 28 00 0c 01 01 01 00 e0 14 03 32 00 0c 01
       01 02
```

// SIFVLAN received a 50-byte broadcast packet whose inner VLAN ID is 13 and outer VLAN ID is 1 on Ethernet 1/1.1.

*Mar 26 17:27:52:671 2008 Sysname SIFVLAN/7/PACKET:

```
Ethernet1/1.1:
IN packet, len 42
   00 e0 14 03 32 00 00 e0 14 03 28 00 08 06 00 01
   08 00 06 04 00 02 00 e0 14 03 28 00 0c 01 01 01
       00 e0 14 03 32 00 0c 01 01 02
```
// SIFVLAN received an untagged 42-byte unicast packet on Ethernet 1/1.1.
VoFR debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

**debugging voice vofr**

Use `debugging voice vofr` to enable VoFR debugging.

Use `undo debugging voice vofr` to disable VoFR debugging.

**Syntax**

```
debugging voice vofr { all | error | event | fax packet { from-dp | from-net | to-dp | to-net } | fsm | info | timer }
undo debugging voice vofr { all | error | event | fax packet { from-dp | from-net | to-dp | to-net } | fsm | info | timer }
```

**Default**

VoFR debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **all**: Specifies all types of debugging for VoFR.
- **error**: Specifies error debugging.
- **event**: Specifies event debugging.
- **fax packet**: Specifies fax packet debugging.
  - **from-dp**: Fax packets from voice interfaces.
  - **from-net**: Fax packets from Frame Relay networks.
  - **to-dp**: Fax packets to voice interfaces.
  - **to-net**: Fax packets to Frame Relay networks.
- **fsm**: Specifies finite state machine debugging.
- **info**: Specifies information debugging.
- **timer**: Specifies timer debugging.

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging voice vofr error` command.
Table 240 Output from the debugging voice vofr error command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid argument</td>
<td>The input parameter is incorrect.</td>
</tr>
<tr>
<td>status error</td>
<td>A call status error.</td>
</tr>
<tr>
<td>attach vofr id error!</td>
<td>VoFR failed to register the VoFR ID with the VIM.</td>
</tr>
<tr>
<td>call info table index error</td>
<td>The call information table index was incorrect.</td>
</tr>
<tr>
<td>not enough memory!</td>
<td>VoFR failed to assign memory because the memory was insufficient.</td>
</tr>
<tr>
<td>create vofr ccb error!</td>
<td>VoFR failed to create a CCB.</td>
</tr>
<tr>
<td>find ccb error!</td>
<td>VoFR failed to find the CCB.</td>
</tr>
<tr>
<td>create ccx error!</td>
<td>VoFR failed to create a CCX.</td>
</tr>
<tr>
<td>assign local call id error!</td>
<td>VoFR failed to assign a local call ID.</td>
</tr>
<tr>
<td>free local call id error!</td>
<td>VoFR failed to release the local call ID.</td>
</tr>
<tr>
<td>assign bandwidth error!</td>
<td>VoFR failed to assign bandwidth.</td>
</tr>
<tr>
<td>free bandwidth error!</td>
<td>VoFR failed to release the bandwidth.</td>
</tr>
<tr>
<td>attach ccx error!</td>
<td>VoFR failed to bind CCB to CCX.</td>
</tr>
<tr>
<td>detach vofr id error!</td>
<td>VoFR failed to deregister the VoFR ID with the VIM.</td>
</tr>
<tr>
<td>find dial peer error!</td>
<td>VoFR failed to find a dial peer.</td>
</tr>
<tr>
<td>send accp message error!</td>
<td>VoFR failed to send an ACCP message.</td>
</tr>
<tr>
<td>send vofr message error!</td>
<td>VoFR failed to send a VoFR message.</td>
</tr>
<tr>
<td>mode error!</td>
<td>The call mode was incorrect.</td>
</tr>
<tr>
<td>get local call id error!</td>
<td>VoFR failed to obtain a local call ID.</td>
</tr>
<tr>
<td>get global call id error!</td>
<td>VoFR failed to obtain a global call ID.</td>
</tr>
<tr>
<td>get message type error!</td>
<td>VoFR failed to obtain the ACCP message type.</td>
</tr>
<tr>
<td>unknown message set!</td>
<td>An unknown message was received.</td>
</tr>
<tr>
<td>unknown fax tone!</td>
<td>An unknown fax tone was received.</td>
</tr>
<tr>
<td>send command to vim error!</td>
<td>An error occurred when a command was sent to the driver.</td>
</tr>
<tr>
<td>recover codec error!</td>
<td>VoFR failed to recover the codec.</td>
</tr>
<tr>
<td>get vpcb error!</td>
<td>VoFR failed to obtain the VPCB pointer. VPCB, a structure type defined by the VIM, is used to record voice interface information, including configuration information and voice interface type.</td>
</tr>
<tr>
<td>get vicb error!</td>
<td>VoFR failed to obtain the VICB pointer. VICB, a structure type defined by the VIM, is used to record the voice channel status in conversation, including transmission direction of trunk signaling and the current channel status.</td>
</tr>
<tr>
<td>unknown timer id!</td>
<td>An unknown timer ID was obtained.</td>
</tr>
<tr>
<td>start fax error!</td>
<td>VoFR failed to start fax.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>delete ccx error!</td>
<td>VoFR failed to delete the CCX.</td>
</tr>
<tr>
<td>vofr local call id is already exist!</td>
<td>The local call ID already existed in the CCB.</td>
</tr>
<tr>
<td>can not assign more vofr call id!</td>
<td>VoFR failed to assign a local call ID.</td>
</tr>
<tr>
<td>get call info tab error!</td>
<td>VoFR failed to obtain a call information table index.</td>
</tr>
<tr>
<td>get switch mode error!</td>
<td>VoFR failed to obtain a switching mode.</td>
</tr>
<tr>
<td>clear fax data in vofr ccb error!</td>
<td>VoFR failed to clear fax data in VoFR CCB.</td>
</tr>
<tr>
<td>release call info table error!</td>
<td>VoFR failed to release call records in the call info</td>
</tr>
<tr>
<td></td>
<td>table.</td>
</tr>
<tr>
<td>send fax tone to net error!</td>
<td>VoFR failed to send fax tones to the network.</td>
</tr>
<tr>
<td>relay fax data to fax module error!</td>
<td>VoFR failed to send fax data to the fax module.</td>
</tr>
<tr>
<td>packet data length error!</td>
<td>A packet data length error occurred.</td>
</tr>
<tr>
<td>packet is dropped by FR!</td>
<td>An error occurred when data was sent to the network</td>
</tr>
<tr>
<td></td>
<td>side.</td>
</tr>
<tr>
<td>write ccb error!</td>
<td>VoFR failed to write data to the CCB.</td>
</tr>
<tr>
<td>create timer T301M error</td>
<td>VoFR failed to create timer T301M.</td>
</tr>
<tr>
<td>channel open failure!</td>
<td>VoFR failed to open a media channel.</td>
</tr>
<tr>
<td>dtmf error!</td>
<td>An invalid DTMF digit was received.</td>
</tr>
<tr>
<td>call info table index error!</td>
<td>Invalid call information table index.</td>
</tr>
</tbody>
</table>

Table 2 describes output fields and messages for the **debugging voice vofr event** command.

**Table 241 Output from the debugging voice vofr event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOFR_STANDARD_SWITCH: send ACCP_INFORMATION[R B START] successfully!</td>
<td>An ACCP_INFORMATION message was sent to the CMC to play ringback tones.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send ACCP_INFORMATION[R B STOP] successfully!</td>
<td>An ACCP_INFORMATION message was sent to the CMC to stop playing ringback tones.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send ACCP_CONNECT successfully!</td>
<td>The ACCP_CONNECT message was sent successfully.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send ACCP_INFORMATION[DTMF ENABLE] successfully!</td>
<td>An ACCP_INFORMATION message was sent to the CMC to start out-of-band DTMF detection.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send ACCP_INFORMATION[DTMF dtmf value] successfully!</td>
<td>An ACCP_INFORMATION message was sent to the CMC. The dtmf value is an out-of-band DTMF digit.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send ACCP_RELEASE successfully!</td>
<td>The ACCP_RELEASE message was sent successfully.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send ACCP_RELEASE_COMPLETE successfully!</td>
<td>The ACCP_RELEASE_COMPLETE message was sent successfully.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send VOFR_SETUP successfully!</td>
<td>The VOFR_SETUP message was sent successfully.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send VOFR_CALL_PROCEEDING successfully!</td>
<td>The VOFR_CALL_PROCEEDING message was sent successfully.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send VOFR_ALERTING successfully!</td>
<td>The VOFR_ALERTING message was sent successfully.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send VOFR_CONNECT successfully!</td>
<td>The VOFR_CONNECT message was sent successfully.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send VOFR_FACILITY successfully!</td>
<td>The VOFR_FACILITY message was sent successfully.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: send VOFR_RELEASE_COMPLETE successfully!</td>
<td>The VOFR_RELEASE_COMPLETE message was sent successfully.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: VOFR_SETUP message received!</td>
<td>A VOFR_SETUP message was received.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: VOFR_CALL_PROCEEDING message received!</td>
<td>A VOFR_CALL_PROCEEDING message was received.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: VOFR_ALERTING message received!</td>
<td>A VOFR_ALERTING message was received.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: VOFR_CONNECT message received!</td>
<td>A VOFR_CONNECT message was received.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: VOFR_RELEASE_COMPLETE message received!</td>
<td>A VOFR_RELEASE_COMPLETE message was received.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: VOFR_FACILITY[DTMF dtmf value] received!</td>
<td>A VOFR_FACILITY message was received from the network side.</td>
</tr>
<tr>
<td>DD: FR channel down.</td>
<td>The RF channel was down.</td>
</tr>
<tr>
<td>Call Protocol: Protocol, FrIfIndex: FR infindex value, FrDlci: dlci number, FrCid: cid number ulStatus: channel status</td>
<td>However, the call protocol, interface index, DLCI number, CID number, and channel status were displayed.</td>
</tr>
<tr>
<td>DD: dlci status change</td>
<td>The RF channel was up.</td>
</tr>
<tr>
<td>FrIfIndex: FR infindex value, FrDlci: dlci number ulStatus: channel status</td>
<td>The interface index, DLCI number, and channel status were displayed.</td>
</tr>
<tr>
<td>remove vofr address information in entity entity index</td>
<td>The VoFR address information in VoFR voice entity entity index was deleted.</td>
</tr>
<tr>
<td></td>
<td>This operation was triggered by removal of a serial interface.</td>
</tr>
<tr>
<td>insert vofr address information in entity entity index</td>
<td>The VoFR address information in VoFR voice entity entity index was recovered.</td>
</tr>
<tr>
<td></td>
<td>This operation was triggered by insertion of a serial interface.</td>
</tr>
</tbody>
</table>
The VoFR address information in VoFR voice entity index was deleted. This operation was triggered by deletion of a serial interface.

The stop-sending-fax-tone event was received from the VIM.

The fax calling tone (CNG) was received from the VIM.

The fax answer tone (CED) was received from the VIM.

The fax-end tone (END) was received from the VIM.

The fax-to-voice switching event started to be processed.

The last T4 data packet was sent to the network.

Table 3 describes output fields and messages for the **debugging voice vofr fax packet from-dp** command.

Table 242 Output from the debugging voice vofr fax packet from-dp command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>len = packet length, time = system time, status = fax status, fax data from dp</td>
<td>The fax data received from the VIM was displayed. The packet length, current system time, status of the fax module, and fax data received from the bottom layer were displayed sequentially.</td>
</tr>
</tbody>
</table>

Table 4 describes output fields and messages for the **debugging voice vofr fax packet from-net** command.

Table 243 Output from the debugging voice vofr fax packet from-net command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>len = packet length, time = system time, status = fax status, fax data from net</td>
<td>The fax data received from the network side was displayed. The packet length, current system time, status of the fax module, and fax data received from the network side were displayed sequentially.</td>
</tr>
</tbody>
</table>

Table 5 describes output fields and messages that require an explanation for the **debugging voice vofr fax packet to-dp** command.

Table 244 Output from the debugging voice vofr fax packet to-dp command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The fax data sent to the bottom layer was displayed. This field is reserved for future use.</td>
<td></td>
</tr>
</tbody>
</table>

Table 6 describes output fields and messages for the **debugging voice vofr fax packet to-net** command.
### Table 245 Output from the debugging voice vofr fax packet to-net command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>len = packet length, time = system time, status = fax status</td>
<td></td>
</tr>
<tr>
<td>fax data to net</td>
<td>The fax data sent to the network side was displayed.</td>
</tr>
<tr>
<td>len = packet length, time = system time, status = fax status</td>
<td></td>
</tr>
<tr>
<td>fax data to net</td>
<td>The packet length, current system time, status of the FAX module, and fax data sent to the network side were displayed sequentially.</td>
</tr>
</tbody>
</table>

### Table 7 describes output fields and messages for the debugging voice vofr fsm command.

### Table 246 Output from the debugging voice vofr fsm command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOFR_STANDARD_SWITCH: call status changed!</td>
<td></td>
</tr>
<tr>
<td>LocalCallID: Local Call ID,</td>
<td></td>
</tr>
<tr>
<td>CurrentStatus: Current call status</td>
<td></td>
</tr>
<tr>
<td>PastStatus: Past call status</td>
<td></td>
</tr>
<tr>
<td>S_IDLE</td>
<td>The CCB was in the idle state.</td>
</tr>
<tr>
<td>S_M_SETUP</td>
<td>Status of the CCB after a VOFR_SETUP message was sent for an outgoing call.</td>
</tr>
<tr>
<td>S_M_CALLPROC</td>
<td>Status of the CCB after a VOFR_CALLPROCEEDING message was received for an outgoing call.</td>
</tr>
<tr>
<td>S_M_ALERTING</td>
<td>Status of the CCB after an ACCP_ALERTING message was sent for an outgoing call.</td>
</tr>
<tr>
<td>S_M_CONNECT</td>
<td>Status of the CCB after a VOFR_CONNECT message was received for an outgoing call.</td>
</tr>
<tr>
<td>S_B_SETUP</td>
<td>Status of the CCB after a VOFR_SETUP message was received for an incoming call.</td>
</tr>
<tr>
<td>S_B_CALLOUT</td>
<td>Status of the CCB after an ACCP_SETUP message was sent to the CMC for an incoming call.</td>
</tr>
<tr>
<td>S_B_SETUPACK</td>
<td>Status of the CCB after an ACCP_SETUP_ACK message was received for an incoming call.</td>
</tr>
<tr>
<td>S_B_ALERTING</td>
<td>Status of the CCB after an ACCP_ALERTING message was received for an incoming call.</td>
</tr>
<tr>
<td>S_B_CONNECT</td>
<td>Status of the CCB after an ACCP_CONNECT message was received for an incoming call.</td>
</tr>
<tr>
<td>VOFR_MODE_VOICE-&gt;VOFR_MODE_SWITCH_RTFAx</td>
<td>The state changed from voice to wait-to-switch-to-fax.</td>
</tr>
<tr>
<td>VOFR_MODE_SWITCH_RTFAx-&gt;VOFR_MODE_VOICE</td>
<td>The state changed from wait-to-switch-to-fax to voice.</td>
</tr>
</tbody>
</table>

### Table 8 describes output fields and messages for the debugging voice vofr info command.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOFR_STANDARD_SWITCH: open channel successfully!</td>
<td>In the Huawei-compatible mode, a media channel was opened successfully.</td>
</tr>
<tr>
<td>fofr fax switch error!</td>
<td>VoFR failed to switch to fax.</td>
</tr>
<tr>
<td>attach vofrid interface vofrid = Local Call ID</td>
<td>The local call ID was registered with the VIM.</td>
</tr>
<tr>
<td>vofrid: Local call id. If vofrid = 255, the local call ID at the local end will be deregistered with the VIM.</td>
<td></td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: assign bandwidth ok! AllocatedBandwidth: ResidualBandwidth:</td>
<td>In the Huawei-compatible mode, VoFR succeeded in assigning bandwidth from FR.</td>
</tr>
<tr>
<td>• AllocatedBandwidth: Bandwidth assigned from FR. • ResidualBandwidth: Available bandwidth on the current link.</td>
<td></td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: the DTMF is DTMF value</td>
<td>In the Huawei-compatible mode, the received out-of-band DTMF digits.</td>
</tr>
<tr>
<td>switching to voice start</td>
<td>Switching from the fax mode to the voice mode.</td>
</tr>
<tr>
<td>callee number = The callee number</td>
<td>Callee number displayed after the terminating side received a VoFR_SETUP message.</td>
</tr>
<tr>
<td>caller number = The caller number</td>
<td>Caller number displayed after the terminating side received a VoFR_SETUP message.</td>
</tr>
<tr>
<td>codec = The first codec type, The send codec type, The third codec type, The fourth codec type</td>
<td>Codec capability set displayed after the terminating side received an VoFR_SETUP message.</td>
</tr>
<tr>
<td>decode type = The decode type</td>
<td>Negotiated codec displayed after the originating side received a VoFR_ALERTING message.</td>
</tr>
<tr>
<td>packet size = size of packet</td>
<td>Packetization period.</td>
</tr>
<tr>
<td>seq = Is sequence valid, timestamp = Is timestamp valid, vad = Is VAD valid</td>
<td>Indicates whether sequence number generation, timestamp, or VAD was enabled on the originating side.</td>
</tr>
<tr>
<td>dtmf = Is outband dtmf valid</td>
<td>Indicates whether the out-of-band DTMF detection was enabled on the originating side.</td>
</tr>
<tr>
<td>entity entity number has been found</td>
<td>The called number matched a voice entity.</td>
</tr>
<tr>
<td>VOFR_STANDARD_SWITCH: find common codec type number common codec type(s).</td>
<td>Common codec types between the local end and the peer end.</td>
</tr>
<tr>
<td>stop ring back success</td>
<td>The CMC was notified to stop playing ringback tones.</td>
</tr>
<tr>
<td>receive vim ced on VOFR_MODE_RTFAX</td>
<td>The CEDs reported by the VIM were received in the fax mode.</td>
</tr>
<tr>
<td>send VICTL_COM_RTFAX_OFF to dsp</td>
<td>DSP was notified to stop faxing.</td>
</tr>
<tr>
<td>delete fax instance</td>
<td>A fax instance was deleted.</td>
</tr>
<tr>
<td>send VICTL_COM_CODEC_UPDATE to dsp</td>
<td>DSP was notified to switch the codec type.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>send VICTL_COM_EC_ON to dsp</td>
<td>DSP was notified to enable the echo cancellation (EC) function.</td>
</tr>
<tr>
<td>send VICTL_COM_EC_OFF to dsp</td>
<td>DSP was notified to disable the EC function.</td>
</tr>
<tr>
<td>send VICTL_COM_VAD_ON to dsp</td>
<td>DSP was notified to enable VAD.</td>
</tr>
<tr>
<td>send VICTL_COM_VAD_OFF to dsp</td>
<td>DSP was notified to disable VAD.</td>
</tr>
<tr>
<td>relay (Number of Fax Data) temporarily saved fax packets to fofr!</td>
<td>Buffered fax packets were sent to the FOFR module for processing.</td>
</tr>
<tr>
<td>get modulation-off again</td>
<td>Modulation-off signals were received from the network side during fax data receiving.</td>
</tr>
<tr>
<td>unrecovered, dis card packet</td>
<td>Two consecutive T.30 fax packets were lost, and the T.30 signaling frame was discarded.</td>
</tr>
<tr>
<td>set unrecovered flag</td>
<td>The corresponding flag was set to True when two consecutive T.30 fax packets were lost.</td>
</tr>
<tr>
<td>need not send tone message</td>
<td>No fax tone was sent to the network side because the send queue length was greater than 0.</td>
</tr>
<tr>
<td>send tone message</td>
<td>Fax tones were sent to the network side.</td>
</tr>
<tr>
<td>create NoCarrier automatically!</td>
<td>No carrier signal.</td>
</tr>
<tr>
<td>send preamble ok</td>
<td>A preamble was sent.</td>
</tr>
<tr>
<td>send last tone message to net</td>
<td>The last fax tone was sent to the network side.</td>
</tr>
<tr>
<td>send last Modulation-On to net</td>
<td>The last modulation-on signal was sent to the network side.</td>
</tr>
<tr>
<td>send last Modulation-Off to net</td>
<td>The last modulation-off signal was sent to the network side.</td>
</tr>
<tr>
<td>send last HDLC-CRC to net</td>
<td>The last HDLC CRC packet was sent to the network side.</td>
</tr>
<tr>
<td>receive single freq from net</td>
<td>A single frequency tone was received from the network side.</td>
</tr>
<tr>
<td>unwanted tone message</td>
<td>A single frequency tone was considered to be an unwanted signal.</td>
</tr>
<tr>
<td>receive preamble from net !</td>
<td>A preamble was received from the network side.</td>
</tr>
<tr>
<td>modulation type = fax_moduration type value</td>
<td>modulation type refers to the fax modulation type.</td>
</tr>
<tr>
<td>no any preamble!</td>
<td>No preamble.</td>
</tr>
<tr>
<td>fax protocol not support!</td>
<td>The fax protocol was not supported.</td>
</tr>
<tr>
<td>limit rate unchanged</td>
<td>There was no need to change the fax rate.</td>
</tr>
<tr>
<td>the callee’s number is callee number</td>
<td>Received callee number.</td>
</tr>
</tbody>
</table>

Table 9 describes output fields and messages for the debugging voice vofr timer command.

**Table 248 Output from the debugging voice vofr timer command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch timer started!</td>
<td>The switch timer was started.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timer VF_FAXTONESENDBYVOICE_ID expire!</td>
<td>The VF_FAXTONESENDBYVOICE_ID timer expired.</td>
</tr>
<tr>
<td>timer VF_VOICETOFAX_ID expire!</td>
<td>The VF_VOICETOFAX_ID timer expired.</td>
</tr>
<tr>
<td>create timer timer name successfully!</td>
<td>VoFR succeeded in creating a timer name.</td>
</tr>
<tr>
<td>delete timer timer name</td>
<td>VoFR succeeded in deleting a timer.</td>
</tr>
<tr>
<td>Process timer name Out</td>
<td>Processing the timer timed out.</td>
</tr>
<tr>
<td>VOFR_T301B</td>
<td>T301B timer with a time length of 240000 ms. VoFR waits for an ACCP_CONNECT message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_T303B</td>
<td>T303B timer with a time length of 60000 ms. VoFR waits for an ACCP_ALERTING message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_TSETUPACK</td>
<td>T_SETTUPACK timer with a time length of 5000 ms. VoFR waits for an ACCP_SETUP_ACK message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_T301M</td>
<td>T301M timer with a time length of 200000 ms. VoFR waits for a VORF_CONNECT message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_T303M</td>
<td>T303M timer with a time length of 60000 ms. VoFR waits for a VORF_CALL_PROCEEDING message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_T310M</td>
<td>T310M timer with a time length of 120000 ms. VoFR waits for a VORF_ALERTING message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_TC301M</td>
<td>TC301M timer with a time length of 180000 ms. VoFR waits for a VORF_CONNECT message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_TC303M</td>
<td>TC303M timer with a time length of 4000 ms. VoFR waits for a VORF_CALL_PROCEEDING or VORF_ALERTING message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_TC308</td>
<td>TC308 timer with a time length of 4000 ms. VoFR waits for a VORF_RELEASE_COMPLETE message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_TC310M</td>
<td>TC310M timer with a time length of 10000 ms. VoFR waits for a VORF_ALERTING or VORF_CONNECT message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_TC313B</td>
<td>TC313B timer with a time length of 4000 ms. VoFR waits for a BSCOT_CONNECTACK message after the timer is started.</td>
</tr>
<tr>
<td>VOFR_TTRUNK</td>
<td>T_TRUNK timer with a default time length of 30000 ms. The timer is started upon on-hook in the FRF.11 trunk mode. The call state is set to idle when the timer expires.</td>
</tr>
</tbody>
</table>
Examples

# Enable debugging for VoFR errors on Router 1. Output similar to the following example is generated when Router 1 originates a call to Router 2 under the following conditions:

- The Huawei-compatible protocol is configured as the call control protocol on Router 1.
- The four codec types configured on Router 1 are different from those configured on Router 2.

<Router1> debugging voice vofr error
*Nov 17 14:19:40:612 2006 Router1 VOFR/7/VOFR:
voice_vofr_error:
  VOFR_STANDARD_SWITCH: look up common codec type error!

// Router 1 and Router 2 do not have common codec types, and the call failed.

The output in the following examples was created under the following conditions:

- The Huawei-compatible protocol is configured as the call control protocol on Router 1.
- After Router 1 establishes a call with Router 2, Router 2 sends a switch-to-fax request to Router 1 and Router 1 sends fax data to Router 2.

# Enable debugging for VoFR events on both Router 1 and Router 2.

<Router1> debugging voice vofr event
<Router2> debugging voice vofr event

- Event debugging on Router 1:
  *Sep 19 11:55:26:815 2006 Router1 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: ACCP_SETUP message received!

    // Router 1 received an ACCP_SETUP message from the CMC.

  *Sep 19 11:55:26:815 2006 Router1 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: send ACCP_SETUP_ACK successfully!

    // An ACCP_SETUP_ACK message was sent to the CMC.

  *Sep 19 11:55:26:816 2006 Router1 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: send VOFR_SETUP successfully!

    // The VOFR_SETUP message was sent to Router 2.

  *Sep 19 11:55:28:29 2006 Router1 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: send ACCP_ALERTING successfully!

    // A ACCP_ALERTING message was sent to the CMC.

  *Sep 19 11:55:28:29 2006 Router1 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: send ACCP_CHANNEL_READY successfully!

  // A VOFR_CALL_PROCEEDING message was received from Router 2.

  *Sep 19 11:55:27:213 2006 Router1 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: VOFR_ALERTING received!

  // A VOFR_ALERTING message was received from Router 2.

  *Sep 19 11:55:28:231 2006 Router1 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: send ACCP_ALERTING successfully!

  // An ACCP_ALERTING message was sent to the CMC.
// An ACCP_CHANNEL_READY message was sent to the CMC.
*Sep 19 11:55:28:433 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    VOFR_STANDARD_SWITCH: ACCP_CHANNEL_READY_ACK message received!

// An ACCP_CHANNEL_READY_ACK was received from the CMC.
*Sep 19 11:55:32:899 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    VOFR_STANDARD_SWITCH: VOFR_CONNECT received!

// A VOFR_CONNECT message was received from Router 2.
*Sep 19 11:55:32:899 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    VOFR_STANDARD_SWITCH: send ACCP_CONNECT successfully!

// An ACCP_CONNECT was sent to the CMC, and the two parties began a conversation.
*Sep 19 11:55:32:899 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    VOFR_STANDARD_SWITCH: send ACCP_INFORMATION[DTMF_ENABLE] successfully!

// An ACCP_INFORMATION message was sent to the CMC to enable out-of-band DTMF detection.
*Sep 19 11:55:35:845 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    VOFR_STANDARD_SWITCH: ACCP_INFORMATION message received!

// An ACCP_INFORMATION message was received from the CMC, and the message carried the out-of-band DTMF digits.
*Sep 19 11:55:35:845 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    VOFR_STANDARD_SWITCH: send VOFR_FACILITY successfully!

// A VOFR_FACILITY message was sent to the network side.
*Sep 19 15:11:20:554 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    receive vim fax tone VI_FAX_TONE_END!

// Fax transmission was completed, and a fax-end tone (END) was received from the driver.
*Sep 19 15:11:20:554 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    fax switch to voice start!

// VoFR switched from fax to voice.
*Sep 19 11:55:47:859 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    VOFR_STANDARD_SWITCH: VOFR_RELEASE_COMPLETE received!

// Router 2 went on-hook first, and Router 1 received an VOFR_RELEASE_COMPLETE message from Router 2.
*Sep 19 11:55:47:859 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
    VOFR_STANDARD_SWITCH: send ACCP_RELEASE successfully!

// An ACCP_RELEASE message was sent to the CMC.
*Sep 19 11:55:47:860 2006 Router1 VOFR/7/VOFR:
voice_vofr_event:
receive accp release complete from cmc!

// An ACCP_RELEASE_COMPLETE message was received from the CMC.

- Event debugging on Router 2:
  * Sep 19 11:52:08:932 2006 Router2 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: VOFR_SETUP received!
    // A VOFR_SETUP message was received from Router 1.
  * Sep 19 11:52:09:347 2006 Router2 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: send VOFR_CALL_PROCEEDING successfully!
    // A VOFR_CALL_PROCEEDING message was sent to Router 1.
  * Sep 19 11:52:09:750 2006 Router2 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: ACCP_SETUP successfully!
    // An ACCP_SETUP message was sent to the CMC.
  * Sep 19 11:52:10:153 2006 Router2 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: ACCP_SETUP_ACK message received!
    // An ACCP_SETUP_ACK message was received from the CMC.
  * Sep 19 11:52:10:768 2006 Router2 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: ACCP_ALERTING message received!
    // An ACCP_ALERTING message was received from the CMC.
  * Sep 19 11:52:11:171 2006 Router2 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: send ACCP_CHANNEL_READY successfully!
    // An ACCP_CHANNEL_READY message was sent to the CMC.
  * Sep 19 11:52:11:373 2006 Router2 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: ACCP_CHANNEL_READY_ACK message received!
    // An ACCP_CHANNEL_READY_ACK message was received from the CMC.
  * Sep 19 11:52:15:14 2006 Router2 VOFR/7/VOFR:
    voice_vofr_event:
    VOFR_STANDARD_SWITCH: ACCP_CONNECT message received!
    // An ACCP_CONNECT message was received from the CMC.
VOFR_STANDARD_SWITCH: send ACCP_INFORMATION[R B STOP] ok!
// An ACCP_INFORMATION message was sent to the CMC to stop playing ringback tones.

*Sep 19 11:52:15:15 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  VOFR_STANDARD_SWITCH: send ACCP_INFORMATION[DTMF ENABLE] successfully!
// An ACCP_INFORMATION message was sent to the CMC to enable out-of-band DTMF detection.

*Sep 19 11:52:15:15 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  VOFR_STANDARD_SWITCH: send VOFR_CONNECT successfully!
// A VOFR_CONNECT message was sent to Router 1.

*Sep 19 11:52:17:961 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  VOFR_STANDARD_SWITCH: VOFR_FACILITY[DTMF 0x8] received!
// A VOFR_FACILITY message was received from Router 1 in conversation, and the message included the out-of-band DTMF digit 0x8.

*Sep 19 11:52:17:961 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  VOFR_STANDARD_SWITCH: send ACCP_INFORMATION[DTMF 8] successfully!
// An ACCP_INFORMATION message was sent to the CMC, and the message included the DTMF digit 8.

*Sep 19 15:06:22:813 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  receive vim fax tone VI_FAX_TONE_CED!
// Router 2 received the fax tone CED reported by the VIM.

*Sep 19 15:08:02:923 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  receive vim fax tone VI_FAX_TONE_END!
// Fax transmission was completed, and a fax-end tone END was received from the driver.

*Sep 19 15:08:02:923 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  fax switch to voice start!
// The state changed from fax to voice.

*Sep 19 11:52:29:974 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  VOFR_STANDARD_SWITCH: ACCP_RELEASE message received!
// Router 2 went on-hook first, and received an ACCP_RELEASE message from the CMC.

*Sep 19 11:52:29:974 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  VOFR_STANDARD_SWITCH: send VOFR_RELEASE_COMPLETE successfully!
// A VOFR_RELEASE_COMPLETE message was sent to Router 1.

*Sep 19 11:52:29:974 2006 Router2 VOFR/7/VOFR:
voice_vofr_event:
  VOFR_STANDARD_SWITCH: send ACCP_RELEASE_COMPLETE successfully!
// An ACCP_RELEASE_COMPLETE message was sent to the CMC.

# On Router 1, enable debugging for VoFR fax packets from voice interfaces.
On Router 1, enable debugging for VoFR fax packets from Frame Relay networks.

On Router 2, enable debugging for VoFR fax packets from Frame Relay networks.

On Router 1, enable debugging for VoFR fax packets to voice interfaces.

Enable debugging for VoFR finite state machines on Router 1 and Router 2.

---

Fax data was received from voice interfaces.
- The data length was 42 bytes.
- The system time was 1367830.
- The fax module was in the FAX_PC_WaitPstnData state.

Fax data was received from the network side.
- The data length was 28 bytes.
- The system time is 1368559.
- The fax module is in the FAX_PC_WaitIpData state.

Fax data was sent to the network side.
- The data length is 28 bytes.
- The system time is 1469278.
- The fax module is in the FAX_PC_WaitPstnData state.

---

The call state of the local CCB changed from S_IDLE to S_M_SETUP, and the local call ID of the CCB was 1.
LocalCallID: 1    CurrentStatus: S_M_CALLPROC,    PastStatus: S_M_SETUP
// The call state of the local CCB changed from S_M_SETUP to S_M_CALLPROC, and the local call ID of the CCB was 1.
*Sep 19 11:55:27:818 2006 Router1 VOFR/7/VOFR:
voice_vofr_fsm:
    VOFR_STANDARD_SWITCH: call status changed!
LocalCallID: 1,    CurrentStatus: S_M_ALERTING,    PastStatus: S_M_CALLPROC
// The call state of the local CCB changed from S_M_CALLPROC to S_M_ALERTING, and the local call ID of the CCB was 1.
*Sep 19 11:55:32:899 2006 Router1 VOFR/7/VOFR:
voice_vofr_fsm:
    VOFR_STANDARD_SWITCH: call status changed!
LocalCallID: 1,    CurrentStatus: S_M_CONNECT,    PastStatus: S_M_ALERTING
// The call status of the local CCB changed from S_M_ALERTING to S_M_CONNECT, and the local call ID of the CCB was 1.
*Sep 19 15:09:40:725 2006 Router1 VOFR/7/VOFR:
voice_vofr_fsm:
    VOFR_MODE_VOICE -> VOFR_MODE_SWITCH_RTFAX
// The state of Router 1 changed from Voice to SWITCH_RTFAX.

- Debugging for VoFR finite state machines on Router 2:
  *Sep 19 11:52:08:932 2006 Router2 VOFR/7/VOFR:
voice_vofr_fsm:
    VOFR_STANDARD_SWITCH: call status changed!
LocalCallID: 1,    CurrentStatus: S_B_SETUP,    PastStatus: S_IDLE
// The call state of the local CCB changed from S_IDLE to S_B_SETUP and the local call ID of the CCB was 1.
*Sep 19 11:52:09:549 2006 Router2 VOFR/7/VOFR:
voice_vofr_fsm:
    VOFR_STANDARD_SWITCH: call status changed!
LocalCallID: 1,    CurrentStatus: S_B_CALLOUT,    PastStatus: S_B_SETUP
// The call state of the local CCB changed from S_B_SETUP to S_B_CALLOUT, and the local call ID of the CCB was 1.
*Sep 19 11:52:09:952 2006 Router2 VOFR/7/VOFR:
voice_vofr_fsm:
    VOFR_STANDARD_SWITCH: call status changed!
LocalCallID: 1,    CurrentStatus: S_B_SETUPACK,    PastStatus: S_B_CALLOUT
// The call state of the local CCB changed from S_B_CALLOUT to S_B_SETUPACK, and the local call ID of the CCB was 1.
*Sep 19 11:52:10:969 2006 Router2 VOFR/7/VOFR:
voice_vofr_fsm:
    VOFR_STANDARD_SWITCH: call status changed!
LocalCallID: 1,    CurrentStatus: S_B_ALERTING,    PastStatus: S_B_SETUPACK
// The call state of the local CCB changed from S_B_SETUPACK to S_B_ALERTING, and the local call ID of the CCB was 1.
LocalCallID: 1,  CurrentStatus: S_B_CONNECT,  PastStatus: S_B_ALERTING
// The call state of the local CCB changed from S_B_ALERTING to S_B_CONNECT, and the local call ID of the CCB was 1.
*Sep 19 15:06:22:813 2006 Router2 VOFR/7/VOFR:
  voice_vofr_fsm:
    VOFR_MODE_VOICE -> VOFR_MODE_SWITCH_RTFAX
// The state of Router 2 changed from Voice to SWITCH_RTFAX.

# Enable debugging for VoFR information on Router 1 and Router 2.
<Router1> debugging voice vofr info
<Router2> debugging voice vofr info

• Debugging for VoFR information on Router 1:
  *Sep 19 11:55:26:815 2006 Router1 VOFR/7/VOFR:
    voice_vofr_info:
      attach vofrid
      interface = subscriber-line5/0
      vofrid = 1
// The local call ID was registered with the VIM. The VoFR ID was 1 because the local call ID was 1.
  *Sep 19 11:55:27:415 2006 Router1 VOFR/7/VOFR:
    voice_vofr_info:
      decode type = 67
// The codec type at the peer end is displayed here. The g729r8 codec was 67.
  *Sep 19 11:55:27:617 2006 Router1 VOFR/7/VOFR:
    voice_vofr_info:
      packet size = 30
// The packetization period of the peer codec is displayed here. The packetization period of the g729r8 codec was 30.
  *Sep 19 11:55:27:818 2006 Router1 VOFR/7/VOFR:
    voice_vofr_info:
      VOFR_STANDARD_SWITCH: assign bandwidth ok!
      AllocatedBandwidth: 10000,  ResidualBandwidth: 43990000
// A bandwidth from FR was successfully assigned. The allocated bandwidth was 10000 bps, and the remaining bandwidth was 43990000 bps.
  *Sep 19 11:55:28:634 2006 Router1 VOFR/7/VOFR:
    voice_vofr_info:
      VOFR_STANDARD_SWITCH: open channel successfully!
// A media channel was successfully opened.
  *Sep 19 11:55:35:845 2006 Router1 VOFR/7/VOFR:
    voice_vofr_info:
      VOFR_STANDARD_SWITCH: the DTMF is 8
// The out-of-band DTMF digit was 8.
  *Sep 19 15:09:40:754 2006 Router1 VOFR/7/VOFR:
    voice_vofr_info:
      relay 1 temporarily saved fax packets to fofr!
// The data buffered in the VoFR CCB was sent to the VoFR module.
  *Sep 19 15:09:45:663 2006 Router1 VOFR/7/VOFR:
voice_vofr_info:
    limit rate unchanged!

    // The current fax rate was allowed, so it was unnecessary to change the fax rate.
*Sep 19 15:09:56:265 2006 Router1 VOFR/7/VOFR:
voice_vofr_info:
    send last Modulation-Off to net

    // A modulation-off signal was sent to the network side.
*Sep 19 15:11:20:554 2006 Router1 VOFR/7/VOFR:
voice_vofr_info:
    send VICTL_COM_RTFAX_OFF to dsp!

    // A signal to the DSP was sent to stop a fax instance.
*Sep 19 15:11:20:554 2006 Router1 VOFR/7/VOFR:
voice_vofr_info:
    delete fax instance!

    // A fax instance was deleted.
*Sep 19 15:11:20:555 2006 Router1 VOFR/7/VOFR:
voice_vofr_info:
    send VICTL_COM_CODEC_UPDATE to dsp!

    // A command was sent to the DSP to switch the codec type.
*Sep 19 15:11:20:555 2006 Router1 VOFR/7/VOFR:
voice_vofr_info:
    send VICTL_COM_EC_ON to dsp!

    // A command was sent to the DSP to enable EC.
*Sep 19 15:11:20:555 2006 Router1 VOFR/7/VOFR:
voice_vofr_info:
    send VICTL_COM_VAD_OFF to dsp!

    // A command was sent to the DSP to disable VAD.
*Sep 19 11:55:47:859 2006 Router1 VOFR/7/VOFR:
voice_vofr_info:
    attach vofrid
    interface = subscriber-line5/0
    vofrid = 255

    // The VoFR ID was deregistered with the VIM. The VoFR ID in the VIM was set to the invalid value 255.

• Debugging for VoFR information on Router 2:
*Sep 19 11:52:08:932 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    callee number = 100

    // The called number was 100.
*Sep 19 11:52:08:933 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    caller number = 200

    // The calling number is 200.
*Sep 19 11:52:08:933 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    codec = 67, 8, 9, 201

483
// The codec capability set was received from Router 1:
  o The g729r8 codec was 67.
  o The g711alaw codec was 8.
  o The g711ulaw codec was 9.
  o The g723r53 codec was 201.

*Sep 19 11:52:08:933 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
  seq = 0,  timestamp = 0,  vad = 0
// seq = 0 means the voice packet carries no sequence number, timestamp = 0 means the voice packet carries no timestamp, and vad = 0 means VAD is disabled.

*Sep 19 11:52:08:933 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
  dtmf = 1
// dtmf = 1 means out-of-band DTMF transmission is enabled.

*Sep 19 11:52:08:933 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
  VOFR_STANDARD_SWITCH: the callee's number is 100.
// The called number was 100.

*Sep 19 11:52:08:933 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
  VOFR_STANDARD_SWITCH: entity 100 has been found!
// The called number matched voice entity 100.

*Sep 19 11:52:10:365 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
  attach vofrid
  interface = subscriber-line5/0
  vofrid = 1
// The local call ID was registered with the VIM. The VoFR ID was 1 because the local call ID is 1.

*Sep 19 11:52:10:566 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
  VOFR_STANDARD_SWITCH: assign bandwidth ok!
  AllocatedBandwidth: 10000,  ResidualBandwidth: 43990000
// A bandwidth from FR was successfully assigned. The allocated bandwidth was 10000 bps, and the remaining bandwidth was 43990000 bps.

*Sep 19 11:52:11:584 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
  VOFR_STANDARD_SWITCH: open channel successfully!
// A media channel was successfully opened.

*Sep 19 11:52:15:15 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
  stop ring back success!
// Ringback tones were successfully stopped.
*Sep 19 15:06:36:673 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    receive preamble from net!
    modulation type = 4

// A preamble was received from the network side. The modulation type of fax data was V29_7200.
*Sep 19 15:06:38:385 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    get modulation-off again!

// A modulation-off signal was received.
*Sep 19 15:08:02:817 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    create NoCarrier automatically!

// There was no carrier signal.
*Sep 19 15:08:02:817 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    no any preamble!

// No preamble was received.
*Sep 19 15:08:02:923 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    send VICTL_COM_RTFAX_OFF to dsp!

// A signal was sent to the DSP to stop fax.
*Sep 19 15:08:02:924 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    delete fax instance!

// A fax instance was deleted.
*Sep 19 15:08:02:924 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    send VICTL_COM_CODEC_UPDATE to dsp!

// A command was sent to the DSP to change the codec type.
*Sep 19 15:08:02:924 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    send VICTL_COM_EC_ON to dsp!

// A command was sent to the DSP to enable EC.
*Sep 19 11:52:29:975 2006 Router2 VOFR/7/VOFR:
voice_vofr_info:
    attach vofrid
    interface = subscriber-line5/0
    vofrid = 255

// The VoFR ID was deregistered with the VIM. The VoFR ID in the VIM was sent to the invalid value 255.

# Enable debugging for VoFR timers on Router 1 and Router 2.
<Router1> debugging voice vofr timer
<Router2> debugging voice vofr timer

* Debugging for VoFR timers on Router 1:
*Sep 19 11:55:26:816 2006 Router1 VOFR/7/VOFR:

  voice_vofr_timer:
  
  VOFR_STANDARD_SWITCH: create timer T303M successfully!

  // The timer T303M was created.
*Sep 19 11:55:26:817 2006 Router1 VOFR/7/VOFR:

  voice_vofr_timer:
  
  VOFR_STANDARD_SWITCH: delete timer T303M.

  // The timer T303M was deleted.
*Sep 19 11:55:27:214 2006 Router1 VOFR/7/VOFR:

  voice_vofr_timer:
  
  VOFR_STANDARD_SWITCH: create timer T310M successfully!

  // The timer T310M was created.
*Sep 19 11:55:27:12 2006 Router1 VOFR/7/VOFR:

  voice_vofr_timer:
  
  VOFR_STANDARD_SWITCH: delete timer T310M.

  // The timer T310M was deleted.
*Sep 19 11:55:32:899 2006 Router1 VOFR/7/VOFR:

  voice_vofr_timer:
  
  VOFR_STANDARD_SWITCH: create timer T301M successfully!

  // The timer T301M was created, with a time length of 200000 ms to wait for a
  // VOFR_CONNECT message.
*Sep 19 11:55:32:899 2006 Router1 VOFR/7/VOFR:

  voice_vofr_timer:
  
  VOFR_STANDARD_SWITCH: delete timer T301M.

  // The timer T301M was deleted.
*Sep 19 15:09:40:725 2006 Router1 VOFR/7/VOFR:

  voice_vofr_timer:
  
  switch timer started!

  // After a fax answer tone (CED) was received from Router 2 (recipient), the switch timer with
  // a time length of 9000 ms was started on Router 1.

• Debugging for VoFR timers on Router 2:
  *Sep 19 11:52:09:549 2006 Router2 VOFR/7/VOFR:

    voice_vofr_timer:
    
    VOFR_STANDARD_SWITCH: create timer T_SetupAck successfully!

    // The T_SetupAck timer was created to wait for an ACCP_SET_UP_ACK message from the CMC.
*Sep 19 11:52:09:750 2006 Router2 VOFR/7/VOFR:

  voice_vofr_timer:
  
  VOFR_STANDARD_SWITCH: delete timer T_SetupAck!

  // The T_SetupAck timer was deleted.
*Sep 19 11:52:09:952 2006 Router2 VOFR/7/VOFR:

  voice_vofr_timer:
  
  VOFR_STANDARD_SWITCH: create timer T303B successfully!

  // The timer T303B was created to wait for an ACCP_ALERTING message from the CMC.
*Sep 19 11:52:10:355 2006 Router2 VOFR/7/VOFR:

  voice_vofr_timer:
VOFR_STANDARD_SWITCH: delete timer T303B!

// The timer T303B was deleted.

*Sep 19 11:52:11:372 2006 Router2 VOFR/7/VOFR:
voice_vofr_timer:
    VOFR_STANDARD_SWITCH: create timer T301B successfully!

// The timer T301B was created.

*Sep 19 11:52:15:14 2006 Router2 VOFR/7/VOFR:
voice_vofr_timer:
    VOFR_STANDARD_SWITCH: delete timer T301B!

// The timer T301B was deleted.

*Sep 19 15:06:22:813 2006 Router2 VOFR/7/VOFR:
voice_vofr_timer:
    switch timer started!

// The switch timer was started.
Voice subscriber line debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

debugging voice cmc

Use `debugging voice cmc` to enable Call Management Center (CMC) debugging.
Use `undo debugging voice cmc` to disable CMC debugging.

**Syntax**

```
debugging voice cmc { all | error | event | fsm | info | timer }
undo debugging voice cmc { all | error | event | fsm | info | timer }
```

**Default**

CMC debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **all**: All types of CMC debugging.
- **error**: CMC error debugging.
- **event**: CMC event debugging.
- **fsm**: CMC FSM debugging.
- **info**: CMC information debugging.
- **timer**: CMC timer debugging.

**Examples**

The output in the following examples was created when the caller 100 calls the callee 400:

```
# Enable CMC event debugging on the originating device.
<Sysname> debugging voice cmc event
Enable CMC EVENT message debugging functions
*Nov 26 09:48:36:343 2006 Sysname CMC/7/VOICE:
LGS -->CMC : ACCP_SETUP               CallID = 0xffffffff LocalID = 0xf
CalledAddr......400
CallerAddr......100
Source IfIndex..0x2c0030
CallInfoTab.....0x0
Q931Info........Invalid
// LGS sent an ACCP_SETU message to CMC.
```
- The call ID is invalid.
- The called address is 400, and the caller address is 100.
- The session source index is 0x2c0030.
- The call information table index is 0x0.
- Q931 transparent transmission is invalid.

*Nov 26 09:48:36:343 2006 Sysname CMC/7/VOICE:*
CMC -->LGS : ACCP_SETUP_ACK  
CallID = 0xc0000  
LocalID = 0xf

*Nov 26 09:48:36:344 2006 Sysname CMC/7/VOICE:*
CMC -->H323 : ACCP_SETUP  
CallID = 0xc0001  
LocalID = 0xffffffff

CalledAddr......400
CallerAddr......100
Source IpAddr...0.0.0.0/0
CallInfoTab.....0x0
Q931Info........Invalid

// After CMS received the ACCP_SETUP message from LGS, CMS sent an ACCP_SETUP message to H.323.

- The local ID is invalid.
- The called address 400, and the caller address is 100.
- The session source address is 0.0.0.0, and the port number is 0.
- The call information table index is 0x0.
- Q931 transparent transmission is invalid.

*Nov 26 09:48:36:345 2006 Sysname CMC/7/VOICE:*
H323-->CMC  : ACCP_SETUP_ACK  
CallID = 0xc0001  
LocalID = 0xb

// CMC received an ACCP_SETUP_ACK message from H.323.

*Nov 26 09:48:36:401 2006 Sysname CMC/7/VOICE:*
H323-->CMC  : ACCP_ALERTING  
CallID = 0xc0001  
LocalID = 0xb  
Target IpAddr...1.1.1.49/1720  
InbandInfo......Invalid  
CallInfoTab.....0xffff  
Q931Info........Valid

// H.323 sent an ACCP_ALERTING message to CMC.

- The destination address is 1.1.1.49, and the port number is 1720.
- The inband information is invalid.
- The call information table index is invalid.
- Q931 transparent transmission is valid.

*Nov 26 09:48:36:401 2006 Sysname CMC/7/VOICE:*
CMC -->LGS  : ACCP_ALERTING  
CallID = 0xc0000  
LocalID = 0xf  
Target IfIndex..0x4fffffff  
InbandInfo......Invalid  
CallInfoTab.....0x0  
Q931Info........Valid

// After CMC received the ACCP_ALERTING message from H.323, CMC sent an ACCP_ALERTING message to LGS.

- The inband information is invalid.
The call information table index is 0x0.

Q931 transparent transmission is valid.

*Nov 26 09:48:43:13 2006 Sysname CMC/7/VOICE:

H323-->CMC : ACCP_INFORMATION         CallID = 0xc0001    LocalID = 0xb
OutBand Dtmf....Disable
DTMF Character..Invalid
EchoCancel IE...Invalid
RingBack IE.....Invalid
HookFlash Evt...Invalid
NTE IE.....Invalid

// H.323 sent an ACCP_INFORMATION message to CMC.

Out-of-band DTMF is disabled.

DTMF characters are invalid.

Echo cancellation is invalid.

Local ringback is invalid.

*Nov 26 09:48:43:13 2006 Sysname CMC/7/VOICE:

CMC -->LGS  : ACCP_INFORMATION         CallID = 0xc0000    LocalID = 0xf
OutBand Dtmf....Disable
DTMF Character..Invalid
EchoCancel IE...Invalid
RingBack IE.....Invalid
HookFlash Evt...Invalid
NTE IE.....Invalid

// After CMC received the ACCP_INFORMATION message from H.323, CMC sent an
ACCP_INFORMATION message to LGS.

Out-of-band DTMF is disabled.

DTMF characters are invalid.

Echo cancellation is invalid.

Local ringback is invalid.

*Nov 26 09:48:43:13 2006 Sysname CMC/7/VOICE:

H323-->CMC  : ACCP_CONNECT             CallID = 0xc0001    LocalID = 0xb
Q931Info........Valid

// After the callee picked up the phone, H.323 sent an ACCP_CONNECT message to CMC. Q931
transparent transmission is valid.

*Nov 26 09:48:43:14 2006 Sysname CMC/7/VOICE:

CMC -->LGS  : ACCP_CONNECT             CallID = 0xc0000    LocalID = 0xf
Q931Info........Valid

// After CMC received the ACCP_INFORMATION message from H.323, CMC sent an
ACCP_CONNECT message to LGS. Q931 transparent transmission is valid.

*Nov 26 09:48:43:15 2006 Sysname CMC/7/VOICE:

LGS -->CMC  : ACCP_CHANNEL_READY       CallID = 0xc0000    LocalID = 0xf
DecodeProtocol..G711A
EncodeProtocol..G711A
Vad Switch......Disable
PayloadSize.....0
// LGS sent an ACCP_CHANNEL_READY to CMC. The codec is G711A. VAD is disabled.
*Nov 26 09:48:43:16 2006 Sysname CMC/7/VOICE:
  CMC -->LGS : ACCP_CHANNEL_READY_ACK  CallID = 0xc0000  LocalID = 0xf
    Media Connect...Success

// LGS received an ACCP_CHANNEL_READY_ACK message from CMC. The media connection was established.
*Nov 26 09:48:43:24 2006 Sysname CMC/7/VOICE:
  H323-->CMC : ACCP_CHANNEL_READY  CallID = 0xc0001  LocalID = 0xb
    DecodeProtocol..G729R8
    EncodeProtocol..G729R8
    Vad Switch......Disable
    PayloadSize.....30

// H.323 sent an ACCP_CHANNEL_READY message to CMC. The codec is G729R8. VAD is disabled.
*Nov 26 09:48:43:25 2006 Sysname CMC/7/VOICE:
  CMC_EVT:IfIndex[2c0030] Send Driver VAD_OFF

// CMC notified the driver to disable VAD.
*Nov 26 09:48:43:25 2006 Sysname CMC/7/VOICE:
  CMC_EVT:IfIndex[2c0030] Send Driver CONNECT 0x[b][1e][b][0][ffffffff]

// CMC notified the driver to connect media.
*Nov 26 09:48:43:42 2006 Sysname CMC/7/VOICE:
  CMC_EVT:CallId[c0001] IfIndex[ffffffff] Media Connect with
    CallId[c0000] IfIndex[2c0030]

*Nov 26 09:48:43:62 2006 Sysname CMC/7/VOICE:
  CMC_EVT:IfIndex[2c0030] Rcv VIM CONNECT_ACK Success

// CMC received an CONNECT_ACK message from VIM.
*Nov 26 09:48:56:15 2006 Sysname CMC/7/VOICE:
  H323-->CMC : ACCP_RELEASE  CallID = 0xc0001  LocalID = 0xb
    ReleaseCause....Normal clearing!
    Q931Info........Valid

// After the callee hung up, H.323 sent an ACCP_RELEASE message to CMC. The reason is a normal disconnection. Q931 transparent transmission is valid.
*Nov 26 09:48:56:15 2006 Sysname CMC/7/VOICE:
  CMC_EVT:IfIndex[2c0030] Send Driver DISCONNECT 0x[ffffffff]

// CMC notified the driver to disconnect media.
*Nov 26 09:48:56:15 2006 Sysname CMC/7/VOICE:
  CMC -->H323 : ACCP_RELEASE_COMPLETE  CallID = 0xc0001  LocalID = 0xb

// H.323 received an ACCP_RELEASE_COMPLETE message from CMC.
*Nov 26 09:48:56:15 2006 Sysname CMC/7/VOICE:
  CMC -->LGS : ACCP_RELEASE  CallID = 0xc0000  LocalID = 0xf
// ReleaseCause....Normal clearing!
Q931Info........Valid

// After CMC received an ACCP_Release message from H.323, CMC sent an ACCP_Release message to LGS. The reason is a normal disconnection. Q931 transparent transmission is valid.

*Nov 26 09:48:56:156 2006 Sysname CMC/7/VOICE:
  LGS --> CMC : ACCP_RELEASE_COMPLETE  CallID = 0xc0000  LocalID = 0xf

// CMC received an ACCP_RELEASE_COMPLETE message from LGS.

# Enable CMC FSM debugging.
<sysname> debugging voice cmc fsm
Enable CMC FSM message debugging functions

*Nov 26 09:51:58:94 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0000] IN_STATE_IDLE ----> IN_STATE_INCOMING_ACK

// The incoming call leg state changed from IN_STATE_IDLE to IN_STATE_INCOMING_ACK.

*Nov 26 09:51:58:95 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0001] OUT_STATE_IDLE ----> OUT_STATE_PRESENT

*Nov 26 09:51:58:96 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0001] OUT_STATE_PRESENT ----> OUT_STATE_OUTGOING_ACK

*Nov 26 09:51:58:259 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0001] OUT_STATE_OUTGOING_ACK ----> OUT_STATE_RECEIVED

*Nov 26 09:51:58:259 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0000] IN_STATE_INCOMING_ACK ----> IN_STATE_DELIVERED

*Nov 26 09:52:01:883 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0001] OUT_STATE_RECEIVED ----> OUT_STATE_ACTIVE

// The outgoing call leg state changed from OUT_STATE_RECEIVED to OUT_STATE_ACTIVE.

*Nov 26 09:52:01:883 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0000] IN_STATE_DELIVERED ----> IN_STATE_ACTIVE

*Nov 26 09:52:06:233 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0001] OUT_STATE_ACTIVE ----> OUT_STATE_IDLE

*Nov 26 09:52:06:234 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0000] IN_STATE_ACTIVE ----> IN_STATE_RELEASE

*Nov 26 09:52:06:235 2006 Sysname CMC/7/VOICE:
  CMC_FSM:CallId[d0000] IN_STATE_RELEASE ----> IN_STATE_IDLE

# Enable CMC information debugging.
<Sysname> debugging voice cmc info
Enable CMC information debugging functions

*May  9 10:01:37:625 2007 Sysname CMC/7/VOICE:
  CMC_INFO: Connect state has already been active in connecting media.

// The call leg has been connected to media.

*May  9 10:01:37:625 2007 Sysname CMC/7/VOICE:
CMC_INFO: Connect state has already been active in connecting media.

*May 9 10:01:37:627 2007 Sysname CMC/7/VOICE:
  CMC_INFO: The brother leg media state is active.

*May 9 10:01:37:635 2007 Sysname CMC/7/VOICE:
  CMC_INFO: Successful to connecting media.

  // The media was successfully connected.

*May 9 10:15:28:266 2007 Sysname CMC/7/VOICE:
  CMC_INFO: Successful to disconnecting media.

  // The media was successfully disconnected after the caller hung up.

*May 9 10:15:28:267 2007 Sysname CMC/7/VOICE:
  CMC_INFO: Get a invalid CMC_SET message(CMC_RELEASE) in service process.

  // CMC received an invalid CMC message in service processing.

# Enable CMC timer debugging.

<Sysname> debugging voice cmc timer
  Enable CMC TIMER message debugging functions

*Nov 26 09:52:26:835 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0000] Create TIMER_INLEG_WAIT_ALERTING Timer
  TimerId       : 272
  TimerLength   : 40000 ms

  // CMC created a timer TIMER_INLEG_WAIT_ALERTING.

*Nov 26 09:52:26:835 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0001] Create TIMER_OUTLEG_WAIT_SETUP_ACK Timer
  TimerId       : 872
  TimerLength   : 5000 ms

*Nov 26 09:52:26:837 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0001] Delete Call Timer
  TimerId       : 872

  // CMC removed the timer with ID 872.

*Nov 26 09:52:26:837 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0001] Create TIMER_OUTLEG_WAIT_ALERTING Timer
  TimerId       : 436
  TimerLength   : 35000 ms

*Nov 26 09:52:26:868 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0001] Delete Call Timer
  TimerId       : 436

*Nov 26 09:52:26:868 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0001] Create TIMER_OUTLEG_WAIT_CONNECT Timer
  TimerId       : 911
  TimerLength   : 120000 ms
*Nov 26 09:52:26:868 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0000] Delete Call Timer
  TimerId     : 272

*Nov 26 09:52:26:868 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0000] Create TIMER_INLEG_WAIT_CONNECT Timer
  TimerId     : 676
  TimerLength : 120000 ms

*Nov 26 09:52:32:33 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0001] Delete Call Timer
  TimerId     : 911

*Nov 26 09:52:32:34 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0000] Delete Call Timer
  TimerId     : 676

*Nov 26 09:52:32:43 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0001] Create TIMER_MDA_CONNECT_ACK Timer
  TimerId     : 891
  TimerLength : 3000 ms

*Nov 26 09:52:32:43 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0001] Delete Media Connect Timer
  TimerId     : 891

*Nov 26 09:52:35:574 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0000] Create TIMER_INLEG_WAIT_RELCOMP Timer
  TimerId     : 539
  TimerLength : 5000 ms

*Nov 26 09:52:35:575 2006 Sysname CMC/7/VOICE:
  CMC_TMR:CallId[e0000] Delete Call Timer
  TimerId     : 539

**debugging voice em**

Use **debugging voice em** to enable E&M debugging.
Use **undo debugging voice em** to disable E&M debugging.

**Syntax**

devbugging voice em { all | error | event | fsm | timer }
undo debugging voice em { all | error | event | fsm | timer }

**Default**

E&M debugging is disabled.

**Views**

User view
Default command level

2: System level

Parameters

- **all**: All types of E&M debugging.
- **error**: E&M error debugging.
- **event**: E&M event debugging.
- **fsm**: E&M FSM debugging.
- **timer**: E&M timer debugging.

Examples

# Enable E&M error debugging. When the cable is disconnected, output similar to the following example is generated:

```
<Sysname> debug voice em error
Enable EM ERROR message debugging functions
*Apr 12 17:51:13:430 2007 Sysname EM/7/VOICE:
    EM_ERROR : Failed to get ViIfIndex.
    // E&M failed to get the interface index.
*Apr 12 17:51:13:431 2007 Sysname EM/7/VOICE:
    EM_ERROR : Null Ccb received Invalid CMC msg.
    // E&M received an invalid CMC message.
```

The output in the following examples was created when the caller calls the number 100:

# Enable E&M event debugging.

```
<Sysname> debug voice em event
Enable EM EVENT message debugging functions
*Aug  2 07:17:52:209 2006 Sysname EM/7/VOICE:
    EM_EVENT [5/0:0.0]: Succeed in creating EMCCB.
    // E&M created EMCCB.
*Aug  2 07:17:52:209 2006 Sysname EM/7/VOICE:
    EM_EVENT [5/0:0.0]: Received ACCP_SETUP from CMC on state EM_IDLE.
    // E&M received an ACCP_SETUP message.
*Aug  2 07:17:52:233 2006 Sysname EM/7/VOICE:
    EM_EVENT [5/0:0.0]: Received VIM_INSTALL_ACK from VIM on state EMCALLER_WAIT_INSTALL_ACK.
    // E&M notified the driver to install DSP resources.
*Aug  2 07:17:52:234 2006 Sysname EM/7/VOICE:
    EM_EVENT [5/0:0.0]: Succeed in sending DTMFDetect off to Driver.
    // E&M disabled DTMF detection.
```

495
Aug 2 07:17:52:234 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/0:0.0]: Succeed in sending AccpSetupAck to CMC.

// E&M sent AccpSetupAck to CMC.
Aug 2 07:17:52:234 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/0:0.0]: Succeed in sending seize digital line signal:0101.

// E&M sent a seize digital line signal.
Aug 2 07:17:52:435 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/1:0.0]: Succeed in creating EMCCB.

// E&M created EMCCB.
Aug 2 07:17:52:435 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/1:0.0]: Received VIM_INPUT_SEIZE from VIM on state EM_IDLE.

// E&M received VIM_INPUT_SEIZE from VIM in state EM_IDLE.
Aug 2 07:17:53:50 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/1:0.0]: Received VIM_INSTALL_ACK from VIM on state EMCALLED_WAIT_RECEIVE_NUMBER.

// E&M received VIM_INSTALL_ACK from VIM in state EMCALLED_WAIT_RECEIVE_NUMBER.
Aug 2 07:17:53:262 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/0:0.0]: Received EM_TIMER_SIGNAL_WAIT on state EMCALLER_WAIT_SEND_NUMBER.

// E&M received EM_TIMER_SIGNAL_WAIT in state EMCALLER_WAIT_SEND_NUMBER.
Aug 2 07:17:53:463 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/1:0.0]: Received VIM_DTMF_IND from VIM on state EMCALLED_RECEIVING_NUMBER.

// E&M received a DTMF digit from VIM in state EMCALLED_RECEIVING_NUMBER.
Aug 2 07:17:53:665 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/1:0.0]: Received DTMF digit 1.

// E&M received DTMF digit 1.
Aug 2 07:17:53:866 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/1:0.0]: Received DTMF 0.

// One DTMF digit (1) was matched.
Aug 2 07:17:53:665 2006 Sysname EM/7/VOICE: 
EM_EVENT [5/1:0.0]: Received phone number 1 partly matched!
// E&M received DTMF digit 0.
*Aug  2 07:17:54:68 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Received phone number 10 partly matched!

// Two DTMF digits (10) were matched.
*Aug  2 07:17:54:78 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Received VIM_DTMF_IND from VIM on state EMCALLED RECEIVING_NUMBER.

// E&M received a DTMF digit from VIM in state EMCALLED RECEIVING_NUMBER.
*Aug  2 07:17:54:279 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Received DTMF 0.

// E&M received DTMF digit 0.
*Aug  2 07:17:54:279 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Received phone number 100 fully matched!

// All the three DTMF digits (100) were matched.
*Aug  2 07:17:54:481 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Succeed in sending AccpSetup to CMC.

// E&M sent AccpSetup to CMC to establish a connection.
*Aug  2 07:17:54:682 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Received ACCP_SETUP_ACK from CMC on state EMCALLED_WAIT SETUP ACK.

// E&M received ACCP SETUP_ACK from CMC in state EMCALLED_WAIT SETUP ACK.
*Aug  2 07:17:54:884 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Succeed in sending DTMFDetect off to Driver.

// E&M disabled DTMF detection.
*Aug  2 07:17:54:884 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Succeed in sending ACCP ALERTING from CMC on state EMCALLED RINGING.

// E&M received ACCP ALERTING from CMC in state EMCALLED RINGING.
*Aug  2 07:17:54:884 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Succeed in sending AccpChannelReady to CMC.

// E&M opened the media channel.
*Aug  2 07:17:55:86 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Succeed in sending ring to Driver.

// E&M sent ringback tones to the driver.
*Aug  2 07:17:55:287 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/1:0.0]: Received AccpChannelReadyAck Msg from CMC.

// E&M received AccpChannelReadyAck from CMC.
*Aug  2 07:17:55:297 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/0:0.0]: Received VIM DTMF_ACK from VIM on state EMCALLER_SENDING NUMBER.

// E&M received VIM DTMF_ACK from VIM in state EMCALLER_SENDING NUMBER.
*Aug  2 07:17:55:509 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/0:0.0]: Succeed in sending AccpChannelReady to CMC.

// E&M notified CMC to open the media channel.
*Aug  2 07:17:55:509 2006 Sysname EM/7/VOICE:
  EM_EVENT [5/0:0.0]: Received AccpChannelReadyAck Msg from CMC.

// E&M received AccpChannelReadyAck from CMC.
// E&M received ACCP_CONNECT from CMC in state EM_CALLING.
// E&M stopped ringing.
// E&M sent a seize digital line signal.
// E&M notified CMC to respond.
// E&M received ACCP_RELEASE from CMC in state EM_CALLER_TALKING.
// E&M disabled DTMF detection.
// E&M sent an idle digital line signal.
// E&M sent a busy tone to the driver.
// E&M sent AccpReleaseComplete to CMC.
// E&M received VIM_INPUT_IDLE from VIM in state EM_CALLER_TALKING.
// E&M received EM_TIMER_SIGNAL_CONFIRM in state EM_CALLER_TALKING.
// E&M sent an idle digital line signal.
// E&M notified CMC to release resources.
EM_EVENT [5/0:0.0]: Received ACCP_RELEASE_COMPLETE from CMC on state EMCALLER_CALLER_ONHOOK.

// E&M received ACCP_RELEASE_COMPLETE from CMC in state EMCALLER_CALLER_ONHOOK.

*Aug 2 07:18:06:445 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/0:0.0]: Succeed in sending idle digital line signal:1101.

// E&M sent an idle digital line signal.

*Aug 2 07:18:06:445 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/0:0.0]: Succeed in sending uninstall to Driver to CMC.

// E&M uninstalled DSP resources.

*Aug 2 07:18:06:646 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/0:0.0]: Succeed in deleting EMCCB.

// E&M deleted EMCCB.

*Aug 2 07:18:06:646 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/1:0.0]: Received VIM_INPUT_IDLE from VIM on state EMCALLED_CALLED_ONHOOK.

// E&M received VIM_INPUT_IDLE from VIM in state EMCALLED_CALLED_ONHOOK.

*Aug 2 07:18:06:848 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/1:0.0]: Received EM_TIMER_SIGNAL_CONFIRM on state EMCALLED_CALLED_ONHOOK.

// E&M received EM_TIMER_SIGNAL_CONFIRM in state EMCALLED_CALLED_ONHOOK.

*Aug 2 07:18:07:49 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/1:0.0]: Succeed in sending Tone Gen Off to Driver.

// E&M sent Tone Gen Off to the driver.

*Aug 2 07:18:07:59 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/1:0.0]: Succeed in sending idle digital line signal:1101.

// E&M sent an idle digital line signal.

*Aug 2 07:18:07:261 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/1:0.0]: Succeed in sending uninstall to Driver to CMC.

// E&M notified the driver to uninstall DSP resources.

*Aug 2 07:18:07:462 2006 Sysname EM/7/VOICE:
   EM_EVENT [5/1:0.0]: Succeed in deleting EMCCB.

// E&M deleted EMCCB.

# Enable E&M FSM debugging.
<Sysname> debug voice em fsm
Enable EM FSM message debugging functions
*Aug 2 09:12:18:106 2006 Sysname EM/7/VOICE:
   EM_FSM [5/0:0.0]: State changed from EM_IDLE to EMCALLER_WAIT_INSTALL_ACK.

// The caller state changed from EM_IDLE to EMCALLER_WAIT_INSTALL_ACK.

*Aug 2 09:12:18:150 2006 Sysname EM/7/VOICE:
   EM_FSM [5/0:0.0]: State changed from EMCALLER_WAIT_INSTALL_ACK to EMCALLER_WAIT_SEND_NUMBER.

// The callers state changed from EMCALLER_WAIT_INSTALL_ACK to EMCALLER_WAIT_SEND_NUMBER.

*Aug 2 09:12:18:204 2006 Sysname EM/7/VOICE:
   EM_FSM [5/1:0.0]: State changed from EM_IDLE to EMCALLED_WAIT_RECEIVE_NUMBER.

// The callee state changed from EM_IDLE to EMCALLED_WAIT_RECEIVE_NUMBER.
EM_FSM [5/1:0.0]: State changed from EMCALLED_WAIT_RECEIVE_NUMBER to
EMCALLED_RECEIVING_NUMBER.

// The callee state changed from EMCALLED_WAIT_RECEIVE_NUMBER to
EMCALLED_RECEIVING_NUMBER.

EM_FSM [5/0:0.0]: State changed from EMCALLER_WAIT_SEND_NUMBER to
EMCALLER_SENDING_NUMBER.

// The caller state changed from EMCALLER_WAIT_SEND_NUMBER to
EMCALLER_SENDING_NUMBER.

EM_FSM [5/1:0.0]: State changed from EMCALLED_RECEIVED_NUMBER to
EMCALLED_WAIT_SETUP_ACK.

// The callee state changed from EMCALLED_RECEIVED_NUMBER to
EMCALLED_WAIT_SETUP_ACK.

EM_FSM [5/0:0.0]: State changed from EMCALLER_SENDING_NUMBER to
EMCALLER_RINGING.

// The caller state changed from EMCALLER_SENDING_NUMBER to
EMCALLER_RINGING.

EM_FSM [5/1:0.0]: State changed from EMCALLED_RINGING to EMCALLED_TALKING.

// The callee state changed from EMCALLED_RINGING to EMCALLED_TALKING.

EM_FSM [5/0:0.0]: State changed from EMCALLER_RINGING to EMCALLER_CALLER_ONHOOK.

// The caller state changed from EMCALLER_RINGING to EMCALLER_CALLER_ONHOOK.

# Enable E&M timer debugging.
<Sysname> debug voice em timer
Enable EM TIMER message debugging functions

EM_TIMER [5/0:0.0]: Created message_wait TimerId [654] which will last 1000ms waiting
for VIM_INSTALL_ACK.

// E&M created the message_wait timer [654] that will wait 1000 ms for VIM_INSTALL_ACK.

EM_TIMER [5/0:0.0]: Deleted TimerId [654].

// E&M deleted the timer [654].

debugging voice ipp

Use **debugging voice ipp** to enable IPP debugging.

Use **undo debugging voice ipp** to disable IPP debugging.

**Syntax**

```
debugging voice ipp { all | cmc | error | rtp-rtcp | socket }
undo debugging voice ipp { all | cmc | error | rtp-rtcp | socket }
```

**Default**

IPP debugging is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **all**: All types of IPP debugging.
- **cmc**: CMC debugging.
- **error**: IPP error debugging.
- **rtp-rtcp**: RTP/RTCP debugging.
- **socket**: Socket debugging.

**Examples**

```
# Enable all types of IPP debugging.
<Sysname> debug voice ipp all
```

depending voice ipp

Use **debugging voice ipp** to enable IPP debugging.

Use **undo debugging voice ipp** to disable IPP debugging.

501
Syntax

disabling voice lgs { all | error | event | fsm | timer }
undo disabling voice lgs { all | error | event | fsm | timer }

Default

LGS debugging is disabled.

Views

User view

Default command level

2: System level

Parameters

all: All types of LGS debugging.
error: LGS error debugging.
event: LGS event debugging.
fsm: LGS FSM debugging.
timer: LGS timer debugging.

Examples

# Enable LGS error debugging. When the caller 200 calls the callee 300 for which the FXS voice
subscriber line has been shut down, output similar to the following example is generated:
<Sysname> debugging voice lgs error
   Enable LGS ERROR message debugging functions
*Feb 28 16:19:01:253 2007 Sysname LGS/7/VOICE:
   LGS_ERROR:[2883633] Create CCB Error When Rcv ACCP_SETUP.
// LGS failed to create CCB upon receiving ACCP_SETUP.
*Feb 28 16:19:01:312 2007 Sysname LGS/7/VOICE:
   LGS_ERROR:[2883633] Send Install VI Command Error.
// LGS failed to send the Install VI command.
The output in the following examples was created when the caller 200 calls the callee 300:
# Enable LGS event debugging.
<Sysname> debugging voice lgs event
   Enable LGS EVENT message debugging functions
*Jul  1 12:46:24:931 2006 Sysname LGS/7/VOICE:
   LGS_EVENT:[2883633] Rcv VIM Message: VIEVT_FXS_OFF_HOOK
// LGS received an off-hook signal from the caller.
*Jul  1 12:46:24:932 2006 Sysname LGS/7/VOICE:
   LGS_EVENT:[2883633] Create CCB, Local Call Ref:3
// LGS created CCB with a local call ref of 3.
*Jul  1 12:46:24:932 2006 Sysname LGS/7/VOICE:
   LGS_EVENT:[2883633] Found CCB for VIM Message VIEVT_FXS_OFF_HOOK, State:LGS_IDLE
// LGS found CCB for VIM Message VIEVT_FXS_OFF_HOOK.
LGS_EVENT:[2883633] Send VI Install Command To Driver

// LGS sent the VI Install command to the driver.
*Jul 1 12:46:24:961 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Rcv VIM Message: VIEVT_COM_INSTALL_ACK: Success

// LGS received VIM message VIEVT_COM_INSTALL_ACK.
*Jul 1 12:46:24:962 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Found CCB for VIM Message VIEVT_COM_INSTALL_ACK, State:FXS_CALLER_INSTALLING

// LGS found CCB for VIM Message VIEVT_COM_INSTALL_ACK in state FXS_CALLER_INSTALLING.
*Jul 1 12:46:25:156 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Send Play Dial Tone Command To Drv, State:FXS_CALLER_INSTALLING

// LGS sent the Play Dial Tone command to the driver in state FXS_CALLER_INSTALLING.
*Jul 1 12:46:25:357 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Send DTMF Detect On Command To Driver

// LGS notified the driver to enable DTMF detection.
*Jul 1 12:46:34:371 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Rcv VIM Message: VIEVT_COM_DTMF_IND: 3

// LGS received called digit 3.
*Jul 1 12:46:34:372 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Found CCB for VIM Message VIEVT_COM_DTMF_IND, State:FXS_CALLER_NUM_RCVING

// LGS found CCB for VIM message VIEVT_COM_DTMF_IND in state FXS_CALLER_NUM_RCVING.
*Jul 1 12:46:34:372 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Send Stop SigTone Command To Driver.

// LGS sent the Stop SigTone command to the driver.
*Jul 1 12:46:34:551 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Rcv VIM Message: VIEVT_COM_DTMF_IND: 0

// LGS received DTMF digit 0 from the driver.
*Jul 1 12:46:34:552 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Found CCB for VIM Message VIEVT_COM_DTMF_IND, State:FXS_CALLER_NUM_RCVING

// LGS found CCB for VIM message VIEVT_COM_DTMF_IND in state FXS_CALLER_NUM_RCVING.
*Jul 1 12:46:34:762 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Rcv VIM Message: VIEVT_COM_DTMF_IND: 0

// LGS received DTMF digit 0 from the driver.
*Jul 1 12:46:34:762 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Found CCB for VIM Message VIEVT_COM_DTMF_IND, State:FXS_CALLER_NUM_RCVING

// LGS received number 300.
*Jul 1 12:46:34:762 2006 Sysname CMC/7/VOICE:
  LGS -->CMC : ACCP_SETUP  
  CalledAddr......300  
  CallerAddr......200  
  Source IfIndex.0xc0031  
  CallInfoTab.....0x0

503
// LGS sent ACCP_SETUP to CMC.
*Jul 1 12:46:34:762 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Send DTMF Detect Off Command To Driver.

// LGS notified the driver to disable DTMF detection.
*Jul 1 12:46:35:364 2006 Sysname CMC/7/VOICE:
CMC -->LGS  : ACCP_SETUP_ACK  CallID = 0x20000  LocalID = 0x3

// CMC sent ACCP_SETUP_ACK to LGS.
*Jul 1 12:46:35:364 2006 Sysname CMC/7/VOICE:
CMC -->LGS  : ACCP_SETUP  CallID = 0x20001  LocalID = 0xffffffff
CalledAddr......300
CallerAddr......200
Source IfIndex..0x2c0031
CallInfoTab.....0x0
Q931Info........Invalid

// CMC sent ACCP_SETUP to LGS.
*Jul 1 12:46:35:566 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] Create CCB, Local Call Ref:4

// LGS created CCB with a local call ref of 4.
*Jul 1 12:46:35:777 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] Send VI Install Command To Driver.

// LGS sent the VI Install command to the driver.
*Jul 1 12:46:36:180 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] Rcv VIM Message: VIEVT_COM_INSTALL_ACK:  Success

// LGS received VIEVT_COM_INSTALL_ACK: Success from the driver.
*Jul 1 12:46:36:382 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] Found CCB for VIM Message VIEVT_COM_INSTALL_ACK,
State:FXS_CALLED_INSTALLING

// LGS found CCB for VIM message VIEVT_COM_INSTALL_ACK in state FXS_CALLED_INSTALLING.
*Jul 1 12:46:36:584 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] Send DTMF Detect Off Command To Driver.

// LGS notified the driver to disable DTMF detection.
*Jul 1 12:46:36:986 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] Send FXS CID Generate Command To Driver,CID type Complex.

// LGS sent the FXS CID Generate command to the driver, with CID type as Complex.
*Jul 1 12:46:36:987 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] Send FXS Start Alerting Command To Driver,
State:FXS_CALLED_INSTALLING

// LGS sent the FXS Start Alerting command to the driver in state FXS_CALLED_INSTALLING.
*Jul 1 12:46:37:188 2006 Sysname CMC/7/VOICE:
LGS -->CMC  : ACCP_SETUP_ACK  CallID = 0x20001  LocalID = 0x4

// LGS sent ACCP_SETUP_ACK to CMC.
*Jul 1 12:46:37:399 2006 Sysname CMC/7/VOICE:
LGS -->CMC  : ACCP_ALERTING  CallID = 0x20001  LocalID = 0x4
// LGS sent ACCP_ALERTING to CMC.
*Jul 1 12:46:37:803 2006 Sysname CMC/7/VOICE:
  CMC -->LGS : ACCP_ALERTING CallID = 0x20000 LocalID = 0x3
  Target IfIndex..0x2c0030
  InbandInfo......Invalid
  CallInfoTab.....0xffff
  Q931Info.......Invalid

// CMC sent ACCP_ALERTING to LGS.
*Jul 1 12:46:38:04 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:Inbandinfo unavail, play ringback on two second later.

// Inband information was invalid. Ringback tones was to be played two seconds later.
*Jul 1 12:46:38:609 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883633] Send Play RingBack Tone Command To Drv, State:FXS_CALLER_RING_BACK

// LGS sent the Play RingBack Tone command to the driver in state FXS_CALLER_RING_BACK.
*Jul 1 12:46:39:22 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883632] Send Play RingBack Tone Command To Drv, State:FXS_CALLER_RING_BACK

// LGS received CID Transmit Success in VIEVT_FXS_CID_ACK.
*Jul 1 12:46:39:223 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883632] Send Play RingBack Tone Command To Drv, State:FXS_CALLER_RING_BACK

*Jul 1 12:46:39:425 2006 Sysname CMC/7/VOICE:
  LGS -->CMC : ACCP_CHANNEL_READY CallID = 0x20001 LocalID = 0x4
  DecodeProtocol..G711A
  EncodeProtocol..G711A
  Vad Switch......Disable
  PayLoadSize.....0

// LGS opened the media channel.
*Jul 1 12:46:39:425 2006 Sysname CMC/7/VOICE:
  CMC -->LGS : ACCP_CHANNEL_READY_ACK CallID = 0x20001 LocalID = 0x4

*Jul 1 12:46:45:561 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883632] Rcv VIM Message: VIEVT_FXS_OFF_HOOK

// The callee received an off-hook signal from the FXS interface.
*Jul 1 12:46:45:561 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883632] Found CCB for VIM Message VIEVT_FXS_OFF_HOOK, State:FXS_CALLED_ALERTING

*Jul 1 12:46:45:561 2006 Sysname LGS/7/VOICE:
  LGS_EVENT:[2883632] Send FXS Stop Alerting Command To Driver, State:FXS_CALLED_ALERTING

// LGS sent the FXS Stop Alerting command to the driver in state FXS_CALLED_ALERTING.
*Jul 1 12:46:45:561 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] MSG_ACCP_CHANNEL_READY message have send.

*Jul 1 12:46:45:561 2006 Sysname CMC/7/VOICE:

    LGS --> CMC : ACCP_CONNECT CallID = 0x20001 LocalID = 0x4
    Q931Info........Invalid

// LGS sent ACCP_CONNECT to CMC.

*Jul 1 12:46:45:562 2006 Sysname CMC/7/VOICE:

    CMC --> LGS : ACCP_CONNECT CallID = 0x20000 LocalID = 0x3
    Q931Info........Invalid

// CMC sent ACCP_CONNECT to LGS.

*Jul 1 12:46:45:764 2006 Sysname LGS/7/VOICE:

LGS_EVENT:[2883633] Send Stop SigTone Command To Driver.

// The caller stopped playing ringback tones.

*Jul 1 12:46:45:965 2006 Sysname CMC/7/VOICE:

    LGS --> CMC : ACCP_CHANNEL_READY CallID = 0x20000 LocalID = 0x3
    DecodeProtocol..G711A
    EncodeProtocol..G711A
    Vad Switch......Disable
    PayloadSize......0

*Jul 1 12:46:45:965 2006 Sysname LGS/7/VOICE:

LGS_EVENT:[2883633] Send Polarity Reverse Command To Driver, State:FXS_CALLER_RING_BACK

// LGS sent the Polarity Reverse command to the driver in state FXS_CALLER_RING_BACK.

*Jul 1 12:46:46:368 2006 Sysname CMC/7/VOICE:

    CMC --> LGS : ACCP_CHANNEL_READY_ACK CallID = 0x20000 LocalID = 0x3
    Media Connect...Success

*Jul 1 12:46:52:532 2006 Sysname LGS/7/VOICE:

LGS_EVENT:[2883632] Rcv VIM Message: VIEVT_FXS_ON_HOOK

// The callee hung up.

*Jul 1 12:46:52:532 2006 Sysname LGS/7/VOICE:

    LGS_EVENT:[2883632] Found CCB for VIM Message VIEVT_FXS_ON_HOOK, State:FXS_CALLED_TALKING

*Jul 1 12:46:52:532 2006 Sysname CMC/7/VOICE:

    LGS --> CMC : ACCP_RELEASE CallID = 0x20001 LocalID = 0x4
    ReleaseCause....Normal onhook by user!
    Q931Info........Invalid

// LGS sent ACCP_RELEASE to CMC.

*Jul 1 12:46:52:533 2006 Sysname LGS/7/VOICE:

    LGS_EVENT:[2883632] Send VI Uninstall Command To Driver, State:LGS_RELEASING

*Jul 1 12:46:52:533 2006 Sysname LGS/7/VOICE:

LGS_EVENT:[2883632] Send Vi Release Command To VIM.

// LGS released callee resources.
*Jul 1 12:46:52:727 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883632] Delete CCB At State:LGS_RELEASING, Local Call Ref:4, CMC ID:131073

*Jul 1 12:46:52:929 2006 Sysname CMC/7/VOICE:
CMC -->LGS : ACCP_RELEASE CallID = 0x20000 LocalID = 0x3
ReleaseCause....Normal onhook by user!
Q931Info.........Invalid

// The caller released local resources upon receiving ACCP_RELEASE.

*Jul 1 12:46:53:130 2006 Sysname CMC/7/VOICE:
LGS -->CMC  : ACCP_RELEASE_COMPLETE CallID = 0x20000 LocalID = 0x3

*Jul 1 12:46:53:332 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Send Polarity Reverse Command To Driver, State:FXS_CALLER_TALKING

*Jul 1 12:46:53:332 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Send Play Busy Tone Command To Drv, State:FXS_CALLER_TALKING

// The caller FXS interface sent busy tones to the driver in state FXS_CALLER_TALKING.

*Jul 1 12:46:53:533 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633]: Do nothing.

*Jul 1 12:46:54:751 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Rcv VIM Message: VIEVT_FXS_ON_HOOK

*Jul 1 12:46:54:752 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Found CCB for VIM Message VIEVT_FXS_ON_HOOK, State:FXS_CALLER_WAITHOOK

*Jul 1 12:46:54:752 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Send Stop SigTone Command To Driver.

// LGS stopped playing busy tones.

*Jul 1 12:46:54:752 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Send VI Uninstall Command To Driver, State:FXS_CALLER_WAITHOOK

// LGS sent the VI Uninstall command to the driver in state FXS_CALLER_WAITHOOK.

*Jul 1 12:46:54:752 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Send Vi Release Command To VIM.

*Jul 1 12:46:54:752 2006 Sysname LGS/7/VOICE:
LGS_EVENT:[2883633] Delete CCB At State:FXS_CALLER_WAITHOOK, Local Call Ref:3, CMC ID:131072

// LGS deleted CCB in state FXS_CALLER_WAITHOOK.

# Enable LGS timer debugging.
<Sysname> debugging voice lgs timer
Enable LGS TIMER message debugging functions

*Jul 1 12:46:24:932 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Create Timer Wait Install ACK Success, State:LGS_IDLE
Timelength:3000 ms

// LGS created the timer Wait Install ACK Success (3000 ms) in state LGS_IDLE.
Jul 1 12:46:24 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Delete Timer Wait Install ACK Success, State:FXS_CALLER_INSTALLING
// LGS deleted the timer Wait Install ACK Success in state XS_CALLER_INSTALLING.

Jul 1 12:46:25 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Create Timer First Dial Success, State:FXS_CALLER_INSTALLING
TimeLength:10000 ms
// LGS created the timer First Dial Success (10000 ms) in state FXS_CALLER_INSTALLING.

Jul 1 12:46:34 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Delete Timer First Dial Success, State:FXS_CALLER_NUM_RCVING
// LGS deleted the timer First Dial Success in state FXS_CALLER_NUM_RCVING.

Jul 1 12:46:34 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Create Timer Dial Interval Success, State:FXS_CALLER_NUM_RCVING
TimeLength:10000 ms
// LGS created the timer Dial Interval Success in state FXS_CALLER_NUM_RCVING.

Jul 1 12:46:35 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883632] Create Timer Wait ACCP_ALERTING Success, State:LGS_IDLE
TimeLength:3000 ms
// LGS created the timer Wait ACCP_ALERTING Success (3000 ms) in state LGS_IDLE.

Jul 1 12:46:36 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883632] Delete Timer Wait Install ACK Success, State:FXS_CALLER_INSTALLING
// LGS deleted the timer Wait Install ACK Success in state FXS_CALLER_INSTALLING.

Jul 1 12:46:37 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883632] Create Timer Wait ACCP_ALERTING Success, State:FXS_CALLER_NUM_RCVING
TimeLength:35000 ms
// LGS created the timer Wait ACCP_ALERTING Success (35000 ms) in state FXS_CALLER_NUM_RCVING.

Jul 1 12:46:38 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883632] Delete Timer Delay Ring Back Success, State:FXS_CALLER_CONNECTING
TimeLength:2000 ms
// LGS created the timer Delay Ring Back Success (2000 ms) in state FXS_CALLER_CONNECTING.
// Ringback expired in state FXS_CALLER_RING_BACK.
*Jul 1 12:46:38:820 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Create Timer Wait ACCP_CONNECT Success, State:FXS_CALLER_RING_BACK
  TimeLength:60000 ms

// LGS created the timer Wait ACCP_CONNECT Success (60000 ms) in state FXS_CALLER_RING_BACK.
*Jul 1 12:46:45:561 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883632] Delete Timer Wait Off Hook Success, State:FXS_CALLED_ALERTING

// LGS deleted the timer Wait Off Hook Success in state FXS_CALLED_ALERTING.
*Jul 1 12:46:45:562 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Delete Timer Wait ACCP_CONNECT Success, State:FXS_CALLER_RING_BACK

// LGS deleted the timer Wait ACCP_CONNECT Success in state FXS_CALLER_RING_BACK.
*Jul 1 12:46:52:532 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883632] Create Timer Wait ACCP_RELEASE_COMPLETE Success,
State:FXS_CALLED_TALKING
  TimeLength:3000 ms

// LGS created the timer Wait ACCP_RELEASE_COMPLETE Success (3000 ms) in state FXS_CALLED_TALKING.
*Jul 1 12:46:52:533 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883632] Delete Timer Wait ACCP_RELEASE_COMPLETE Success,
State:LGS_RELEASING

// LGS deleted the timer Wait ACCP_RELEASE_COMPLETE Success in state LGS_RELEASING.
*Jul 1 12:46:53:745 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Create Timer Wait On Hook Success, State:FXS_CALLER_TALKING
  TimeLength:900000 ms

// LGS created the timer Wait On Hook Success (900000 ms) in state FXS_CALLER_TALKING.
*Jul 1 12:46:54:752 2006 Sysname LGS/7/VOICE:
LGS_TIMER:[2883633] Delete Timer Wait On Hook Success, State:FXS_CALLER_WAITHOOK

// LGS deleted the timer Wait On Hook Success in state FXS_CALLER_WAITHOOK.

# Enable LGS FSM debugging.
<Sysname> debugging voice lgs fsm
Enable LGS FSM message debugging functions

*Jul 1 12:46:24:392 2006 Sysname LGS/7/VOICE:
LGS_FSM:[2883633] State Change From LGS_IDLE To FXS_CALLER_INSTALLING

// State changed from LGS_IDLE to FXS_CALLER_INSTALLING.
*Jul 1 12:46:25:559 2006 Sysname LGS/7/VOICE:
LGS_FSM:[2883633] State Change From FXS_CALLER_INSTALLING To FXS_CALLER_NUM_RCVING

// State changed from FXS_CALLER_INSTALLING to FXS_CALLER_NUM_RCVING.
*Jul 1 12:46:35:163 2006 Sysname LGS/7/VOICE:
LGS_FSM:[2883633] State Change From FXS_CALLER_NUM_RCVING To FXS_CALLER_CONNECTING

// State changed from FXS_CALLER_NUM_RCVING to FXS_CALLER_CONNECTING.
*Jul 1 12:46:36:180 2006 Sysname LGS/7/VOICE:
LGS_FSM:[2883632] State Change From LGS_IDLE To FXS_CALLED_INSTALLING

// State changed from LGS_IDLE to FXS_CALLED_INSTALLING.
debugging voice r2

Use `debugging voice r2` to enable R2 debugging.

Use `undo debugging voice r2` to disable R2 debugging.

Syntax

```
display debugging voice r2 { all | error | event | fsm | info | timer | trace } [ subscriber-line line-number ]
undo debugging voice r2 { all | error | event | fsm | info | timer | trace } [ subscriber-line line-number ]
```

Default

R2 debugging is disabled.

Views

User view

Default command level

2: System level

Parameters

- `all`: All types of R2 debugging.
- `error`: R2 error debugging.
- `event`: R2 event debugging.
- `fsm`: R2 FSM debugging.
- `info`: R2 information debugging.
- `timer`: R2 timer debugging.
- `trace`: R2 trace debugging.
- `subscriber-line line-number`: R2 debugging for the specified interface.
# Enable R2 error debugging. When the caller configured with R2 signaling calls the callee not configured with R2 signaling, output similar to the following example is generated:

```bash
<Sysname> debugging voice r2 error
*Apr 13 10:42:28:687 2007 Sysname R2/7/VOICE:
R2_ERROR[D-Invalid]:Allocate VI error.
// R2 failed to allocate VI.
*Apr 13 10:42:28:688 2007 Sysname R2/7/VOICE:
R2_ERROR[D-Invalid]:ACCP message error.
// An ACCP message error occurred.
```

# Enable R2 event debugging. When the caller 200 calls the callee 500300, output similar to the following example is generated:

```bash
<Sysname> debugging voice r2 event
Enable voice r2 event debugging functions.
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[D-Invalid]:Receive CMC ACCP_SETUP message:
    CMD ID    : 0x1d0001        Caller Addr: 200
    Local ID  : 0xffffffff      Called Addr: 500300
// R2 received ACCP_SETUP from CMC.
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[O-5/0:0.1]:Send to DRV [VICTL_COM_INSTALL],install DSP.
// R2 sent the Install DSP command to the driver.
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[O-5/0:0.1]:Receive DRV [VI_INSTALL_ACK] message:
    VIIFINDEX : 0x2c3302      Result : SUCCESS
// R2 received VI_INSTALL_ACK from the driver.
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[O-5/0:0.1]:Send to CMC ACCP_SETUP_ACK message:
    CMC ID : 0x1d0001      Local ID : 0x1e
// R2 sent ACCP_SETUP_ACK to CMC.
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Send to DRV VICTL_COM_MFC_DETECT_ON.
// R2 sent line signal [Seizure] 0001 to the driver.
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Receive DRV line signal [0001] on state [DL_IDLE].
// R2 received DRV line signal [0001] in state [DL_IDLE].
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Send to DRV VICTL_COM_MFC_DETECT_ON.
// R2 sent DRV VICTL_COM_MFC_DETECT_ON to the driver.
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Receive DRV MFC [DigitalSignal] -- [3].
// R2 sent MFC number 3 and signal 3 to the driver.
*Dec 22 10:39:11:375 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Receive DRV MFC [DigitalSignal] -- [3].
```
State : [REG_GROUP_A] Stage : [REG_WAIT_CALLEDNUMBER]

// R2 received MFC number 3 in REG_GROUP_A state at REG_WAIT_CALLEDNUMBER stage.
*Dec 22 10:39:12:285 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Send to DRV [ReqNextCalledNum] -- [1]!

// R2 sent ReqNextCalledNum 1 to the driver.
*Dec 22 10:39:12:436 2006 Sysname R2/7/VOICE:
R2_EVENT[O-5/0:0.1]:Send to DRV MFC [MFC_SEND_STOP] -- [0].

// R2 sent MFC_SEND_STOP signal 0 to the driver.
*Dec 22 10:39:12:436 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Receive DRV MFC [MFC_SEND_STOP] -- [0].

// R2 received MFC_SEND_STOP signal 0 from the driver.
*Dec 22 10:39:12:598 2006 Sysname R2/7/VOICE:
R2_EVENT[O-5/0:0.1]:Send to DRV [VICTL_COM_MFC_DETECT_OFF], stop MFC detect.

// R2 sent Stop MFC Detect to the driver.
*Dec 22 10:39:17:263 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Send to DRV VICTL_COM_TONE_GEN_ON, play ringback.

// R2 sent the Play Ringback command to the driver.
*Dec 22 10:39:23:86 2006 Sysname R2/7/VOICE:
R2_EVENT[I-5/1:0.1]:Send to DRV [VICTL_COM_TONE_GEN_OFF], stop ring.

// R2 sent the Stop Ring command to the driver.

The output in the following examples was created when the caller calls the callee:

# Enable R2 FSM debugging.
<Sysname> debugging voice r2 fsm
   Enable voice r2 fsm debugging functions.
*Dec 22 10:35:48:665 2006 Sysname R2/7/VOICE:
R2_FSM[O-5/0:0.1]:State change from [CTL_INIT] to [CTL_WAIT].

// State changed from [CTL_INIT] to [CTL_WAIT].
*Dec 22 10:35:48:665 2006 Sysname R2/7/VOICE:
R2_FSM[O-5/0:0.1]:State change from [DL_IDLE] to [DL_TAKE].!

// State changed from [DL_IDLE] to [DL_TAKE]
*Dec 22 10:35:48:696 2006 Sysname R2/7/VOICE:
R2_FSM[I-5/1:0.1]:State change from [REG_IDLE] to [REG_GROUP_A].

// State changed from [REG_IDLE] to [REG_GROUP_A].
*Dec 22 10:35:48:696 2006 Sysname R2/7/VOICE:
R2_FSM[I-5/1:0.1]:Set stage [REG_WAIT_CALLEDNUMBER].

// Set the stage to REG_WAIT_CALLEDNUMBER.

# Enable R2 information debugging.
<Sysname> debugging voice r2 info
   Enable voice r2 info debugging functions.
*Dec 22 10:36:33:815 2006 Sysname R2/7/VOICE:
R2_INFO[O-5/0:0.1]:CCB create OK, LocalID = 0x1a.

// R2 created CCB with local ID 0x1a.
*Dec 22 10:36:36:310 2006 Sysname R2/7/VOICE:
R2_INFO[I-5/1:0.1]:MFC complete.

// MFC was completed.
-Dec 22 10:36:44:811 2006 Sysname R2/7/VOICE:
R2_INFO[D-5/1:0.1]:Deleted Ccb at state CTL_RELEASEING:
  CMC ID : 0x1a0000      Local ID : 0x1b

// R2 deleted CCB in state CTL_RELEASEING.

# Enable R2 timer debugging.
<.Sysname> debugging voice r2 timer
  Enable voice r2 timer debugging functions.
-Dec 22 10:34:47:225 2006 Sysname R2/7/VOICE:
R2_TIMER[O-5/0:0.1]:Start [CTL_VI_INSTALL_TIMER].
  timerID = [496]    timerlen = [3000]

// R2 enabled the CTL_VI_INSTALL_TIMER timer.
-Dec 22 10:34:47:255 2006 Sysname R2/7/VOICE:
R2_TIMER[D-5/0:0.1]:Delete CTL timer ok,timerID=[496].

// R2 deleted the CTL timer.

# Enable R2 trace debugging.
<Sysname> debugging voice r2 trace
  Enable voice r2 trace debugging functions.
-Dec 22 10:37:56:706 2006 Sysname R2/7/VOICE:
R2_TRACE[D-5/0:0.1]:DL process CTL event [CTL_DL_TKO_SEIZURE] on state [DL_IDLE].

// DL processed the CTL event CTL_DL_TKO_SEIZURE.
-Dec 22 10:37:56:709 2006 Sysname R2/7/VOICE:
R2_TRACE[O-5/1:0.1]:CTL process DL event [DL_CTL_TKI_SEIZURE]!

// DL processed the CTL event DL_CTL_TKI_SEIZURE.

**debugging voice statistics**

Use **debugging voice statistics** to enable voice statistics debugging.
Use **undo debugging voice statistics** to disable voice statistics debugging.

**Syntax**

```
    debugging voice statistics { all | error | info }
    undo debugging voice statistics { all | error | info }
```

**Default**

Voice statistics debugging is disabled.

**Views**

User view

**Default command level**

1: Monitor level

**Parameters**

- **all**: All types of voice statistics debugging.
**error:** Voice statistics error debugging.

**info:** Voice statistics internal information debugging.

**Examples**

# Enable all types of voice statistics debugging on the originating device. When Telephone 1 (with number 100 and IP address 100.1.1.189) calls Telephone 2 (with number 200 and IP address 100.1.1.188), output similar to the following example is generated:

```
<Sysname> debug voice statistics all
<Sysname>
#Jun  5 11:03:52:798 2008 Sysname STAT/4/VOICE:
1.3.6.1.2.1.10.21.2.0.2<dialCtlPeerCallSetup> The trap already send.
*Jun  5 11:03:52:798 2008 Sysname STAT/7/VOICE:
  STAT_INFO: Add a new item(0) to call active table successfully.
// After Telephone 2 was picked up, a new entry (0) was added to the call active table.
*Jun  5 11:03:56:210 2008 Sysname STAT/7/VOICE:
  STAT_INFO: Add a new item(1) to call active table successfully.
// A new call active table entry (1) was added.
*Jun  5 11:03:56:210 2008 Sysname STAT/7/VOICE:
  STAT_INFO: The trap switch of entity 200 is off.
<Sysname>
*Jun  5 11:04:12:527 2008 Sysname STAT/7/VOICE:
  STAT_INFO: Remove the item(1) from call active table successfully.
// When Telephone 2 disconnected, the call active table entry (1) was deleted.
*Jun  5 11:04:14:498 2008 Sysname STAT/7/VOICE:
  STAT_INFO: Remove the item(0) from call active table successfully.
// The call active table entry (0) was deleted.
```
VRRP debugging commands

The output description tables in this document only contain fields and messages that require an explanation.

The term "router" in this chapter refers to both routers and Layer 3 switches.

IPv4 VRRP debugging commands

debugging vrrp error

Use `debugging vrrp error` to enable IPv4 VRRP error debugging.
Use `undo debugging vrrp error` to disable IPv4 VRRP error debugging.

Syntax

```
  debugging vrrp error  
  undo debugging vrrp error
```

**Default**
IPv4 VRRP error debugging is disabled.

**Views**
User view

**Default command level**
1. Monitor level

**Examples**

```
# Enable IPv4 VRRP error debugging. (VRRP operates in standard mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp error
The VRRP packet is illegal  
  // An illegal VRRP packet was received.

# Enable IPv4 VRRP error debugging. (VRRP operates in load balancing mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp error
*Apr 27 21:57:23:763 2000 Sysname VRRP/7/DebugError:  
The VRRP packet is illegal  
  // An illegal VRRP packet was received.
```
**debugging vrrp event**

Use `debugging vrrp event` to enable IPv4 VRRP event debugging.

Use `undo debugging vrrp event` to disable IPv4 VRRP event debugging.

**Syntax**

```
debugging vrrp event
undo debugging vrrp event
```

**Default**

IPv4 VRRP event debugging is disabled.

**Views**

User view

**Default command level**

1. Monitor level

**Usage guidelines**

Table 1 describes output fields and messages for the `debugging vrrp event` command.

**Table 249 Output from the debugging vrrp event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Status of track object `track-entry-number` changed to state | Track object state:  
  * positive.  
  * negative.  
  * invalid.  |
| IPv4 `interface-name` | Forwarder `vrid.vfid` : event | Event that occurred on VF `vfid` in IPv4 VRRP group `vrid` on interface `interface-name`:  
  * Active timer created.  
  * Tick timer created.  
  * VF instance created.  
  * VF instance deleted.  
  * Active timer deleted.  
  * Offer timer created.  
  * Offer timer deleted.  
  * Offer timer expired.  
  * Tick timer deleted.  
  * Redirect timer expired.  
  * Time-out timer expired.  
  * Forward information updated.  
  * Forward information deleted.  
  * Virtual MAC `mac-address` added.  
  * Virtual MAC `mac-address` deleted. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 interface-name</td>
<td>Event that occurred on IPv4 VRRP group vrid on interface interface-name:</td>
</tr>
<tr>
<td></td>
<td>• Adver timer created.</td>
</tr>
<tr>
<td></td>
<td>• Adver timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Hold timer created.</td>
</tr>
<tr>
<td></td>
<td>• Hold timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Hold timer expired.</td>
</tr>
<tr>
<td></td>
<td>• VF-learning timer created.</td>
</tr>
<tr>
<td>Virtual Router vrid</td>
<td>• VF-learning timer deleted.</td>
</tr>
<tr>
<td>event</td>
<td>• VF-learning timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Request timer created.</td>
</tr>
<tr>
<td></td>
<td>• Request timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Request timer expired.</td>
</tr>
</tbody>
</table>

Examples

# Enable IPv4 VRRP event debugging. (VRRP operates in standard mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp event

# Create IPv4 VRRP group 1 on interface Ethernet 1/1, and assign virtual IP address 3.1.1.3 to the VRRP group.
<Sysname> system-view
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] vrrp vrid 1 virtual-ip 3.1.1.3
IPv4 Ethernet1/1 | Virtual Router 1 : Hold timer created
// The Hold timer was created.

IPv4 Ethernet1/1 | Virtual Router 1 : Hold timer expired
// The Hold timer expired.

IPv4 Ethernet1/1 | Virtual Router 1 : Hold timer deleted
// The Hold timer was deleted.

IPv4 Ethernet1/1 | Virtual Router 1 : Adver timer created
// The Adver timer was created.

# Enable IPv4 VRRP event debugging. (VRRP operates in load balancing mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp event

# Specify the load balancing mode for IPv4 VRRP.
<Sysname> system-view
Syntax

```
debugging vrrp packet [ interface interface-type interface-number [ vrid virtual-router-id ] ]
undo debugging vrrp packet [ interface interface-type interface-number [ vrid virtual-router-id ] ]
```

Default
IPv4 VRRP packet debugging is disabled.

Views
User view

Default command level
1. Monitor level

Parameters

- `interface interface-type interface-number`: Specifies an interface by its type and number.
- `vrid virtual-router-id`: Specifies a VRRP group by its ID in the range of 1 to 255.

Usage guidelines

Disabling packet debugging globally does not disable packet debugging for a specific interface or VRRP group. If you enable packet debugging on an interface or VRRP group and disable packet debugging globally, the system still outputs debugging information for that interface or VRRP group.

Table 2 describes output fields and messages for the `debugging vrrp packet` command.

**Table 250 Output from the debugging vrrp packet command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sent <code>message-type</code> message from <code>interface-name</code></td>
<td>Message types include Advertisement, Request, Reply, and Release.</td>
</tr>
<tr>
<td>Received <code>message-type</code> message from <code>ip-address</code> on <code>interface-name</code></td>
<td>Message types include Advertisement, Request, Reply, and Release.</td>
</tr>
<tr>
<td>VRID</td>
<td>VRRP group ID.</td>
</tr>
<tr>
<td>Pri</td>
<td>Priority of the master or the VF.</td>
</tr>
<tr>
<td>Adver timer</td>
<td>VRRP advertisement sending interval in seconds.</td>
</tr>
<tr>
<td>Weight</td>
<td>Weight of the VF.</td>
</tr>
<tr>
<td>VMAC</td>
<td>Virtual MAC address assigned by the master.</td>
</tr>
<tr>
<td>Forwarder number</td>
<td>VF information.</td>
</tr>
<tr>
<td>Owner ID</td>
<td>Real MAC address of the VF owner.</td>
</tr>
<tr>
<td>MAC</td>
<td>Real MAC address of the master.</td>
</tr>
<tr>
<td>Timer</td>
<td>Remaining time of the redirect timer/timeout timer, in seconds.</td>
</tr>
<tr>
<td>IP</td>
<td>Real IP address of the master.</td>
</tr>
</tbody>
</table>

Examples

```
# Enable IPv4 VRRP packet debugging. (VRRP operates in standard mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp packet
```
# Create IPv4 VRRP group 1 on interface Ethernet 1/1, assign virtual IP address 3.1.1.3 to the VRRP group, and configure the router priority as 100.

<Sysname> system-view
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] vrrp vrid 1 virtual-ip 3.1.1.3
*Oct 15 11:24:00:172 2008 Sysname VRRP/7/DebugPacket:
  Sent Advertisement message from Ethernet1/1
  VRID: 1  Pri: 100  Adver timer: 1 secs
// Ethernet 1/1 sent an advertisement with VRID 1, router priority 100, and a sending interval of 1 second.

# Enable IPv4 VRRP packet debugging. (VRRP operates in load balancing mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp packet

# Specify the load balancing mode for IPv4 VRRP.
<Sysname> system-view
[Sysname] vrrp mode load-balance

# Create IPv4 VRRP group 1 on interface Ethernet 1/1, and assign virtual IP address 3.1.1.3 to the VRRP group.
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] vrrp vrid 1 virtual-ip 3.1.1.3
  Sent Request message from Ethernet1/1
  VRID: 1  MAC: 0000-5e01-1101  IP: 3.1.1.1
// A request with MAC address 0000-5e01-1101 and IP address 3.1.1.1 was sent from Ethernet 1/1.
  Sent Advertisement message from Ethernet1/1
  VRID: 1  Pri: 100  Adver timer: 1 secs
// Ethernet 1/1 sent an advertisement with VRID 1, router priority 100, and a sending interval of 1 second.
  Sent Advertisement message from Ethernet1/1
  VRID: 1  Pri: 100  Adver timer: 1 secs
  Weight: 255
  Forwarder 1:
    Pri: 255  Timer: 600/1800 secs  Owner ID: 0000-5e01-1101
// Ethernet 1/1 sent an advertisement.

debugging vrrp state

Use debugging vrrp state to enable IPv4 VRRP state debugging.
Use undo debugging vrrp state to disable IPv4 VRRP state debugging.

Syntax

debugging vrrp state
undo debugging vrrp state
Default

IPv4 VRRP state debugging is disabled.

Views

User view

Default command level

1. Monitor level

Usage guidelines

Table 3 describes output fields and messages for the `debugging vrrp state` command.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 interface-name</td>
<td>Virtual Router vrid : state1 --&gt; state2</td>
</tr>
<tr>
<td></td>
<td>IPv4 VRRP group vrid on interface interface-name transitioned from state1 to state2.</td>
</tr>
<tr>
<td></td>
<td>VRRP group states:</td>
</tr>
<tr>
<td></td>
<td>• Created.</td>
</tr>
<tr>
<td></td>
<td>• Inactive.</td>
</tr>
<tr>
<td></td>
<td>• Initialize.</td>
</tr>
<tr>
<td></td>
<td>• Backup.</td>
</tr>
<tr>
<td></td>
<td>• Master.</td>
</tr>
<tr>
<td></td>
<td>• Deleted.</td>
</tr>
</tbody>
</table>

IPv4 interface-name | Virtual Router vrid : state1 --> state2 reason: reason |
<p>| | IPv4 VRRP group vrid on interface interface-name transitioned from state1 to state2 due to reason. |
| | VRRP group states: |
| | • Created. |
| | • Inactive. |
| | • Initialize. |
| | • Backup. |
| | • Master. |
| | • Deleted. |
| | State transition reasons: |
| | • Timer expired. |
| | • VRRP packet received. |
| | • Adding virtual MAC address failed. |
| | • Interface event received. |
| | • IP address deleted. |
| | • Insufficient hardware resources. |
| | • Illegal IP addresses added. |
| | • The status of the tracked object changed. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 interface-name</td>
<td>VF vfid in IPv4 VRRP group vrid on interface interface-name transitioned from state1 to state2 due to reason.</td>
</tr>
<tr>
<td>vrid</td>
<td>VF states:</td>
</tr>
<tr>
<td>vfid</td>
<td>• Initialize.</td>
</tr>
<tr>
<td></td>
<td>• Listening.</td>
</tr>
<tr>
<td></td>
<td>• Active.</td>
</tr>
<tr>
<td>state1 --&gt; state2</td>
<td>State transition reasons:</td>
</tr>
<tr>
<td>reason: reason</td>
<td>• Weight changed.</td>
</tr>
<tr>
<td></td>
<td>• Adding virtual MAC address failed.</td>
</tr>
<tr>
<td></td>
<td>• Conceded—The AVF dropped the forwarding capability by sending a packet with priority 0.</td>
</tr>
<tr>
<td></td>
<td>• Learnt from Advertisement.</td>
</tr>
<tr>
<td></td>
<td>• Reply received.</td>
</tr>
<tr>
<td></td>
<td>• Release received.</td>
</tr>
<tr>
<td></td>
<td>• Active timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Time-out timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Self-allocated—The master assigned a virtual MAC address to itself.</td>
</tr>
<tr>
<td></td>
<td>• VRRP down.</td>
</tr>
<tr>
<td></td>
<td>• Take over—The LVF became the AVF when it received no advertisement within 3 times the advertisement sending interval.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable IPv4 VRRP state debugging. (VRRP operates in standard mode.)

```
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp state
```

# Create IPv4 VRRP group 1 on interface Ethernet 1/1, and assign virtual IP address 3.1.1.3 to the VRRP group.

```
<Sysname> system-view
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] vrrp vrid 1 virtual-ip 3.1.1.3
```


// VRRP group 1 transitioned from Created to Initialize.

// VRRP group 1 transitioned from Initialize to Backup.

// VRRP group 1 transitioned from Backup to Master when it received no advertisement within 3 times the advertisement sending interval.
# Enable IPv4 VRRP state debugging. (VRRP operates in load balancing mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp state

# Specify the load balancing mode for IPv4 VRRP.
<Sysname> system-view
[Sysname] vrrp mode load-balance

# Create IPv4 VRRP group 1 on interface Ethernet 1/1, and assign virtual IP address 3.1.1.3 to the VRRP group.
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] vrrp vrid 1 virtual-ip 3.1.1.3

IPv4 Ethernet1/1 | Virtual Router 1 : Created --> Initialize

// VRRP group 1 transitioned from Created to Initialize.
IPv4 Ethernet1/1 | Virtual Router 1 : Initialize --> Backup

// VRRP group 1 transitioned from Initialize to Backup.
IPv4 Ethernet1/1 | Virtual Router 1 : Backup --> Master reason: Timer expired

// VRRP group 1 transitioned from Backup to Master when it received no advertisement within 3 times the advertisement sending interval.
IPv4 Ethernet1/1 | Forwarder 1.1 : Initialize --> Active reason: Self-allocated

// The master in VRRP group 1 assigned a virtual MAC address to itself and created VF 1. The VF transitioned from Initialize to Active.

IPv6 VRRP debugging commands

debugging vrrp ipv6 error

Use debugging vrrp ipv6 error to enable IPv6 VRRP error debugging.

Use undo debugging vrrp ipv6 error to disable IPv6 VRRP error debugging.

Syntax

debugging vrrp ipv6 error
undo debugging ipv6 vrrp error

Default

IPv6 VRRP error debugging is disabled.

Views

User view

Default command level

1. Monitor level
Examples

# Enable IPv6 VRRP error debugging. (VRRP operates in standard mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp ipv6 error
*Apr 27 21:56:57:300 2000 Sysname VRRP/7/DebugError:
The VRRP Packet is illegal
// An illegal VRRP packet was received.

# Enable IPv6 VRRP error debugging. (VRRP operates in load balancing mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp ipv6 error

# Specify the load balancing mode for IPv6 VRRP.
<Sysname> system-view
[Sysname] vrrp mode load-balance
*Apr 27 21:56:57:300 2000 Sysname VRRP/7/DebugError:
The VRRP Packet is illegal
// An illegal VRRP packet was received.

devugging vrrp ipv6 event

Use **debugging vrrp ipv6 event** to enable IPv6 VRRP event debugging.

Use **undo debugging vrrp ipv6 event** to disable IPv6 VRRP event debugging.

**Syntax**

```
devugging vrrp ipv6 event
undo devugging vrrp ipv6 event
```

**Default**

IPv6 VRRP event debugging is disabled.

**Views**

User view

**Default command level**

1. Monitor level

**Usage guidelines**

Table 4 describes output fields and messages for the **debugging vrrp ipv6 event** command.

**Table 252 Output from the debugging vrrp ipv6 event command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status of VRRP IPv6 track object</td>
<td>Track object state:</td>
</tr>
<tr>
<td>track-entry-number changed to state</td>
<td>• positive.</td>
</tr>
<tr>
<td></td>
<td>• negative.</td>
</tr>
<tr>
<td></td>
<td>• invalid.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>IPv6 interface-name</td>
<td>Event that occurred on VF vfid in IPv6 VRRP group vrid on interface interface-name:</td>
</tr>
<tr>
<td></td>
<td>• Active timer created.</td>
</tr>
<tr>
<td></td>
<td>• Tick timer created.</td>
</tr>
<tr>
<td></td>
<td>• VF instance created.</td>
</tr>
<tr>
<td></td>
<td>• VF instance deleted.</td>
</tr>
<tr>
<td></td>
<td>• Active timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Offer timer created.</td>
</tr>
<tr>
<td></td>
<td>• Offer timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Offer timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Tick timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Redirect timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Time-out timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Forward information updated.</td>
</tr>
<tr>
<td></td>
<td>• Forward information deleted.</td>
</tr>
<tr>
<td></td>
<td>• Virtual MAC mac-address added.</td>
</tr>
<tr>
<td></td>
<td>• Virtual MAC mac-address deleted.</td>
</tr>
<tr>
<td>IPv6 interface-name</td>
<td>Event that occurred on IPv6 VRRP group vrid on interface interface-name:</td>
</tr>
<tr>
<td></td>
<td>• Adver timer created.</td>
</tr>
<tr>
<td></td>
<td>• Adver timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Hold timer created.</td>
</tr>
<tr>
<td></td>
<td>• Hold timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Hold timer expired.</td>
</tr>
<tr>
<td></td>
<td>• VF-learning timer created.</td>
</tr>
<tr>
<td></td>
<td>• VF-learning timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• VF-learning timer expired.</td>
</tr>
<tr>
<td></td>
<td>• Request timer created.</td>
</tr>
<tr>
<td></td>
<td>• Request timer deleted.</td>
</tr>
<tr>
<td></td>
<td>• Request timer expired.</td>
</tr>
</tbody>
</table>

**Examples**

# Enable IPv6 VRRP event debugging. (VRRP operates in standard mode.)
```bash
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp ipv6 event
```

# Create IPv6 VRRP group 1 on interface Ethernet 1/1, and assign virtual IPv6 address fe80::10 to the VRRP group.
```bash
<Sysname> system-view
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] vrrp ipv6 vrid 1 virtual-ip fe80::10 link-local
```
Oct 15 13:12:08:94 2008 Sysname VRRP/7/DebugEvent:
IPv6 Ethernet1/1 | Virtual Router 1 : Hold timer created

// The Hold timer was created.

Oct 15 13:12:10:313 2008 Sysname VRRP/7/DebugEvent:
IPv6 Ethernet1/1 | Virtual Router 1 : Hold timer expired

// The Hold timer expired.

Oct 15 13:12:10:313 2008 Sysname VRRP/7/DebugEvent:
IPv6 Ethernet1/1 | Virtual Router 1 : Hold timer deleted

// The Hold timer was deleted.

Oct 15 13:12:10:328 2008 Sysname VRRP/7/DebugEvent:
IPv6 Ethernet1/1 | Virtual Router 1 : Adver timer created

// The Adver timer was created.

Oct 15 13:12:10:328 2008 Sysname VRRP/7/DebugEvent:
Send unsolicited ND.

// An unsolicited ND packet was sent.

# Enable IPv6 VRRP event debugging. (VRRP operates in load balancing mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp ipv6 event

# Specify the load balancing mode for IPv6 VRRP.
<Sysname> system-view
[Sysname] vrrp mode load-balance

# Create IPv6 VRRP group 1 on interface Ethernet 1/1, and assign virtual IPv6 address fe80::10 to the VRRP group.
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] vrrp ipv6 vrid 1 virtual-ip fe80::10 link-local

IPv6 Ethernet1/1 | Virtual Router 1 : Hold timer created

// The Hold timer was created.

IPv6 Ethernet1/1 | Virtual Router 1 : Adver timer created

// The Adver timer was created.

IPv6 Ethernet1/1 | Virtual Router 1 : VF-learning timer created

// The VF-learning timer was created.

IPv6 Ethernet1/1 | Virtual Router 1 : VF-learning timer deleted

// VF-learning timer expired.

IPv6 Ethernet1/1 | Virtual Router 1 : VF-learning timer expired

// VF-learning timer expired.

IPv6 Ethernet1/1 | Virtual Router 1 : VF-learning timer deleted
debugging vrrp ipv6 packet

Use `debugging vrrp ipv6 packet` to enable IPv6 VRRP packet debugging. Use `undo debugging vrrp ipv6 packet` to disable IPv6 VRRP packet debugging.

Syntax

```
debugging vrrp ipv6 packet [ interface interface-type interface-number [ vrid virtual-router-id ] ]
undo debugging vrrp ipv6 packet [ interface interface-type interface-number [ vrid virtual-router-id ] ]
```

Default

IPv6 VRRP packet debugging is disabled.

Views

User view

Default command level

1. Monitor level

Parameters

- `interface interface-type interface-number`: Specifies an interface by its type and number.
- `vrid virtual-router-id`: Specifies a VRRP group by its ID in the range of 1 to 255.

Usage guidelines

Disabling packet debugging globally does not disable packet debugging for a specific interface or VRRP group. If you enable packet debugging on an interface or VRRP group and disable packet debugging globally, the system still outputs debugging information of that interface or VRRP group.

Table 5 describes output fields and messages for the `debugging vrrp ipv6 packet` command.

527
### Table 253 Output from the debugging vrrp ipv6 packet command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sent message-type</td>
<td>The message types include Advertisement, Request, Reply, and Release.</td>
</tr>
<tr>
<td>message from</td>
<td></td>
</tr>
<tr>
<td>interface-name</td>
<td></td>
</tr>
<tr>
<td>Received message-type</td>
<td>The message types include Advertisement, Request, Reply, and Release.</td>
</tr>
<tr>
<td>message from</td>
<td></td>
</tr>
<tr>
<td>ip-address on</td>
<td></td>
</tr>
<tr>
<td>interface-name</td>
<td></td>
</tr>
<tr>
<td>VRID</td>
<td>VRRP group ID.</td>
</tr>
<tr>
<td>Pri</td>
<td>Priority of the master or the VF.</td>
</tr>
<tr>
<td>Adver timer</td>
<td>VRRP advertisement sending interval in centiseconds.</td>
</tr>
<tr>
<td>Weight</td>
<td>Weight of the VF.</td>
</tr>
<tr>
<td>VMAC</td>
<td>Virtual MAC address assigned by the master.</td>
</tr>
<tr>
<td>Forwarder</td>
<td>VF information.</td>
</tr>
<tr>
<td>Owner ID</td>
<td>Real MAC address of the VF owner.</td>
</tr>
<tr>
<td>Timer</td>
<td>Remaining time of the redirect timer/timeout timer, in seconds.</td>
</tr>
<tr>
<td>MAC</td>
<td>Real MAC address of the master.</td>
</tr>
<tr>
<td>IP</td>
<td>Real IP address of the master.</td>
</tr>
</tbody>
</table>

### Examples

#### # Enable IPv6 VRRP packet debugging. (VRRP operates in standard mode.)

```bash
<Sysname> terminal debugging  
<Sysname> terminal monitor  
<Sysname> debugging vrrp ipv6 packet  
```

#### # Create IPv6 VRRP group 1 on interface Ethernet 1/1, and assign virtual IPv6 address FE80::1 to the VRRP group.

```bash
<Sysname> system-view  
[Sysname] interface ethernet 1/1  
[Sysname-Ethernet1/1] vrrp ipv6 vrid 1 virtual-ip fe80::1 link-local  
```

**Oct 15 11:54:08:828 2008 Sysname VRRP/7/DebugPacket:**  
Sent Advertisement message from Ethernet1/1  
VRID: 1 Pri: 255 Adver timer: 100 centisecs  

// Ethernet 1/1 sent an advertisement.

#### # Enable IPv6 VRRP packet debugging. (VRRP operates in load balancing mode.)

```bash
<Sysname> terminal debugging  
<Sysname> terminal monitor  
<Sysname> debugging vrrp ipv6 packet  
```

#### # Specify the load balancing mode for IPv6 VRRP.

```bash
<Sysname> system-view  
[Sysname] vrrp mode load-balance  
```

#### # Create IPv6 VRRP group 1 on interface Ethernet 1/1, and assign virtual IPv6 address FE80::10 to the VRRP group.

```bash
[Sysname] interface Ethernet 1/1  
[Sysname-Ethernet1/1] vrrp ipv6 vrid 1 virtual-ip fe80::10 link-local  
```

528
debugging vrrp ipv6 state

Use `debugging vrrp ipv6 state` to enable IPv6 VRRP state debugging.

Use `undo debugging vrrp ipv6 state` to disable IPv6 VRRP state debugging.

Syntax

```
debugging vrrp ipv6 state
undo debugging vrrp ipv6 state
```

Default
IPv6 VRRP state debugging is disabled.

Views
User view

Default command level
1. Monitor level

Usage guidelines

Table 6 describes output fields and messages for the `debugging vrrp ipv6 state` command.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 <code>interface-name</code></td>
<td>IPv6 VRRP group vrid on interface <code>interface-name</code> transitioned from <code>state1</code> to <code>state2</code>.</td>
</tr>
<tr>
<td>Virtual Router <code>vrid</code></td>
<td>VRRP group states:</td>
</tr>
<tr>
<td><code>state1</code></td>
<td>• Created.</td>
</tr>
<tr>
<td><code>state2</code></td>
<td>• Inactive.</td>
</tr>
<tr>
<td><code>reason</code></td>
<td>• Initialize.</td>
</tr>
<tr>
<td><code>state1</code></td>
<td>• Backup.</td>
</tr>
<tr>
<td><code>state2</code></td>
<td>• Master.</td>
</tr>
<tr>
<td><code>reason</code></td>
<td>• Deleted.</td>
</tr>
</tbody>
</table>

IPv6 `interface-name` | Virtual Router `vrid` : `state1` -> `state2` transitioned from `state1` to `state2` due to `reason`. VRRP group states: • Created. • Inactive. • Initialize. • Backup. • Master. • Deleted. State transition reasons: • Timer expired. • VRRP packet received. • Adding virtual MAC address failed. • Interface event received. • IP address deleted. • Insufficient hardware resources. • Illegal IP addresses added. • The status of the tracked object changed.
VF vfid in IPv6 VRRP group vrid on interface interface-name transitioned from state1 to state2 due to reason.

VF states:
- Initialize.
- Listening.
- Active.

State transition reasons:
- Weight changed.
- Adding virtual MAC address failed.
- Conceded—The AVF dropped the forwarding capability by sending a packet with priority 0.
- Learnt from Advertisement.
- Reply received.
- Release received.
- Active timer expired.
- Time-out timer expired.
- Self-allocated—The master assigned a virtual MAC address to itself.
- VRRP down.
- Take over—The LVF became the AVF when it received no advertisement within 3 times the advertisement sending interval.

Examples

# Enable IPv6 VRRP state debugging. (VRRP operates in standard mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp ipv6 state

# Create IPv6 VRRP group 1 on interface Ethernet 1/1, and assign virtual IPv6 address FE80::10 to the VRRP group.
<Sysname> system-view
<Sysname-Ethernet1/1] vrrp ipv6 vrid 1 virtual-ip fe80::10 link-local
IPv6 Ethernet1/1 | Virtual Router 1 : Created --> Initialize

// VRRP group 1 transitioned from Created to Initialize.
IPv6 Ethernet1/1 | Virtual Router 1 : Initialize --> Backup

// VRRP group 1 transitioned from Initialize to Backup.
IPv6 Ethernet1/1 | Virtual Router 1 : Backup --> Master reason: Timer expired
// VRRP group 1 transitioned from Backup to Master when it received no advertisement within 3 times the advertisement sending interval.

# Enable IPv6 VRRP state debugging. (VRRP operates in load balancing mode.)
<Sysname> terminal debugging
<Sysname> terminal monitor
<Sysname> debugging vrrp ipv6 state

# Specify the load balancing mode for IPv6 VRRP.
<Sysname> system-view
[Sysname] vrrp mode load-balance

# Create IPv6 VRRP group 1 on interface Ethernet 1/1, and assign virtual IPv6 address FE80::10 to the VRRP group.
[Sysname] interface ethernet 1/1
[Sysname-Ethernet1/1] vrrp ipv6 vrid 1 virtual-ip fe80::10 link-local

   IPv6 Ethernet1/1 | Virtual Router 1 : Created --> Initialize

// VRRP group 1 transitioned from Created to Initialize.
*Oct 15 13:40:40:141 2008 Sysname VRRP/7/DebugState:
   IPv6 Ethernet1/1 | Virtual Router 1 : Initialize --> Backup

// VRRP group 1 transitioned from Initialize to Backup.
   IPv6 Ethernet1/1 | Virtual Router 1 : Backup --> Master reason: Timer expired

// VRRP group 1 transitioned from Backup to Master when it received no advertisement within 3 times the advertisement sending interval.
   IPv6 Ethernet1/1 | Forwarder 1.1 : Initialize --> Active reason: Self-allocated

// The master in VRRP group 1 assigned a virtual MAC address to itself and created VF 1. The VF transitioned from Initialize to Active.
WEB filtering debugging commands

debugging firewall http url-filter host

Use **debugging firewall http url-filter host** to enable debugging for URL address filtering.
Use **undo debugging firewall http url-filter host** to disable debugging for URL address filtering.

**Syntax**

```
debugging firewall http url-filter host { event | packet }
undo debugging firewall http url-filter host { event | packet }
```

**Default**

Debugging for URL address filtering is disabled.

**Views**

User view

**Default command level**

2: System level

**Parameters**

- **event**: Specifies event debugging.
- **packet**: Specifies packet debugging.

**Examples**

# Enable packet debugging for URL address filtering. Output similar to the following example is generated when a client on the private network accesses www.webfilter.com under the following conditions:

- The URL address filtering function is configured on the device.

```
<Sysname> debugging firewall http url-filter host packet
*Feb 24 10:33:58:500 2007 Sysname DPALG/7/debug:
URL host filter dropped the packet.
```

// A request packet was dropped by the URL address filter.

---

debugging firewall http url-filter parameter

Use **debugging firewall http url-filter parameter** to enable debugging for URL parameter filtering.
Use **undo debugging firewall http url-filter parameter** to disable debugging for URL parameter filtering.

**Syntax**

```
debugging firewall http url-filter parameter { event | packet }
undo debugging firewall http url-filter parameter { event | packet }
```

533
Default

Debugging for URL parameter filtering is disabled.

Views

User view

Default command level

2: System level

Parameters

- `event`: Specifies event debugging.
- `packet`: Specifies packet debugging.

Examples

# Enable packet debugging for URL parameter filtering. Output similar to the following example is generated when a client on the private network accesses www.webfilter.com/parares.html?id=1%20and%20id%20=2 under the following conditions:

- The URL parameter filtering function is configured on the device.

<Sysname> debugging firewall http url-filter parameter packet
Feb 24 10:33:58:500 2007 Sysname DPALG/7/debug:
URL host parameter dropped the packet.
// A request packet was dropped by the URL parameter filter.

debugging firewall http java-blocking

Use `debugging firewall http java-blocking` to enable debugging for Java applet blocking.

Use `undo debugging firewall http java-blocking` to disable debugging for Java applet blocking.

Syntax

```
debugging firewall http java-blocking { event | packet }
undo debugging firewall http java-blocking { event | packet }
```

Default

Debugging for Java applet blocking is disabled.

Views

User view

Default command level

2: System level

Parameters

- `event`: Specifies event debugging.
- `packet`: Specifies packet debugging.
Examples

# Enable packet debugging for Java applet blocking. Output similar to the following example is generated when a client on the private network accesses www.webfilter.com/alcsfnow.class under the following conditions:
- The Java applet blocking function is configured on the device.
- The function denies Web requests containing Java applet.

<Sysname> debugging firewall http activex-blocking packet
*Feb 24 10:33:58:500 2007 Sysname DPALG/7/debug:
Java applet blocking replaced the suffix.
// A file suffix in the packet was replaced.

debugging firewall http activex-blocking

Use debugging firewall http activex-blocking to enable debugging for ActiveX blocking.
Use undo debugging firewall http activex-blocking to disable debugging for ActiveX blocking.

Syntax

debugging firewall http activex-blocking { event | packet }
undo debugging firewall http activex-blocking { event | packet }

Default

Debugging for ActiveX blocking is disabled.

Views

User view

Default command level

2: System level

Parameters

event: Specifies event debugging.
packet: Specifies packet debugging.

Examples

# Enable packet debugging for ActiveX blocking. Output similar to the following example is generated when a client on the private network accesses 1.1.1.1/test.ocx under the following conditions:
- The ActiveX blocking function is configured on the device.
- The function permits Web requests containing ActiveX.

<Sysname> debugging firewall http activex-blocking packet
*Feb 24 10:33:58:500 2007 Sysname DPALG/7/debug:
ActiveX blocking didn’t replace the suffix, because 1.1.1.1 is permitted by ACL 2000.