



## Debug Commands

---

The commands in this chapter apply to the Cisco MDS 9000 Family of multilayer directors and fabric switches. All debug commands are issued in EXEC mode and are shown here in alphabetical order. For more information, refer to the *Cisco MDS 9000 Family Troubleshooting Guide* and the *Cisco MDS 9000 Family System Messages Guide*.

Using the CLI, you can enable debugging modes for each switch feature and view a real-time updated activity log of the control protocol exchanges. Each log entry is time-stamped and listed in chronological order. Access to the debug feature can be limited through the CLI roles mechanism and can be partitioned on a per-role basis.

- [debug all, page 6-3](#)
- [debug bootvar, page 6-4](#)
- [debug callhome, page 6-5](#)
- [debug cdp, page 6-6](#)
- [debug core, page 6-7](#)
- [debug ethport, page 6-8](#)
- [debug exceptionlog, page 6-9](#)
- [debug fc2, page 6-10](#)
- [debug fcc, page 6-12](#)
- [debug fcdomain, page 6-14](#)
- [debug fcfwd, page 6-16](#)
- [debug fcns, page 6-17](#)
- [debug fcs, page 6-18](#)
- [debug flogi, page 6-19](#)
- [debug fspf, page 6-21](#)
- [debug hardware, page 6-23](#)
- [debug idehsd, page 6-24](#)
- [debug ipconf, page 6-25](#)
- [debug ipfc, page 6-26](#)
- [debug ips, page 6-27](#)
- [debug logfile, page 6-29](#)
- [debug mcast, page 6-30](#)

- [debug mip, page 6-32](#)
- [debug module, page 6-33](#)
- [debug ntp, page 6-34](#)
- [debug platform, page 6-35](#)
- [debug port, page 6-36](#)
- [debug port-channel, page 6-38](#)
- [debug qos, page 6-39](#)
- [debug rd-reg, page 6-40](#)
- [debug rdl, page 6-41](#)
- [debug rib, page 6-42](#)
- [debug rscn, page 6-43](#)
- [debug scsi-target, page 6-44](#)
- [debug security, page 6-45](#)
- [debug sensor, page 6-46](#)
- [debug snmp, page 6-47](#)
- [debug span, page 6-49](#)
- [debug system, page 6-50](#)
- [debug tcap, page 6-51](#)
- [debug tlport, page 6-52](#)
- [debug ttyd, page 6-53](#)
- [debug vni, page 6-54](#)
- [debug vrrp, page 6-55](#)
- [debug vsan, page 6-56](#)
- [debug wr-reg, page 6-57](#)
- [debug wwn, page 6-58](#)
- [debug xbar, page 6-59](#)
- [debug xbc, page 6-60](#)
- [debug zone, page 6-61](#)

# debug all

To enable debugging for all features on the switch, use the **debug all** command in EXEC mode. You can disable this command and turn off all debugging by using the **no** form of this command.

**debug all**

**no debug all**

---

**Syntax Description** This command has no arguments or keywords.

---

**Defaults** None.

---

**Command Modes** EXEC mode.

---

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

---

**Usage Guidelines** None.

---

**Examples** The following example displays the system output when the **debug all** command is issued:

```
switch# debug all
```

# debug bootvar

To enable debugging for boot variables, use the **debug bootvar** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug bootvar all | errors | events | info | pss**

Syntax Description		
	<b>all</b>	Enables all boot variable debug options.
	<b>errors</b>	Enables debugging for boot variable errors.
	<b>events</b>	Enables debugging for boot variable events.
	<b>info</b>	Enables debugging for boot variable information.
	<b>pss</b>	Enables debugging for boot variable PSS operations.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug bootvar all** command is issued:

```
switch# debug bootvar all
```

# debug callhome

To enable debugging for the Call Home function, use the **debug callhome** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug callhome all | events | mts**

Syntax Description	all	Enables debugging for all Call Home features.
	events	Enables debugging for all Call Home events.
	mts	Enables debugging for all Call Home tx/rx packets of MTS

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug callhome events** command is issued:

```
switch# debug callhome events
Apr  8 13:09:37 callhome: Src: 0x00000501/4065 Dst: 0x00000501/66 ID: 0x0004FA
05 Size: 252 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x0004FA05 HA_SEQNO:
0x00000000 TS: 0x8657D581CAE REJ:0
Apr  8 13:09:37 callhome: 00 00 00 00 64 00 00 00 00 00 00 00 00 00 00 00
Apr  8 13:09:37 callhome: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr  8 13:09:37 callhome: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
...
```

The following example displays the system output when the **debug callhome mts** command is issued:

```
switch# debug callhome mts
Apr  8 13:09:42 callhome: Src: 0x00000501/4067 Dst: 0x00000501/66 ID: 0x0004FA
0D Size: 252 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x0004FA0D HA_SEQNO:
0x00000000 TS: 0x86708AFE37B REJ:0
Apr  8 13:09:42 callhome: 00 00 00 00 06 00 00 00 00 00 00 00 00 00 00 00
Apr  8 13:09:42 callhome: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr  8 13:09:42 callhome: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
...
Apr  8 13:09:42 callhome: Src: 0x00000501/4067 Dst: 0x00000501/66 ID: 0x0004FA
10 Size: 252 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x0004FA10 HA_SEQNO:
0x00000000 TS: 0x86708D6A974 REJ:0
Apr  8 13:09:42 callhome: 00 00 00 00 05 00 00 00 00 00 00 00 00 00 00 00
Apr  8 13:09:42 callhome: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr  8 13:09:42 callhome: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
...
```

# debug cdp

To enable debugging for the CDP function, use the **debug cdp** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug cdp
  [all] |
  [errors] |
  [events ( mts | packets | pss ) ( interface gigabitethernet slot-port | mgmt 0 ) ]
```

Syntax Description		
<b>all</b>		Enables debugging for all CDP features.
<b>errors</b>		Enables debugging for CDP error conditions.
<b>events</b>		Enables debugging for CDP events.
<b>mts</b>		Enables debugging for CDP tx/rx MTS packets.
<b>packets</b>		Enables debugging for CDP tx/rx CDP packets.
<b>pss</b>		Enables debugging for all PSS related CDP events.
<b>interface</b>		Specifies debugging for the specified interface.
<b>gigabitethernet slot-port</b>		Specifies the Gigabit Ethernet interface slot and port.
<b>mgmt 0</b>		Specifies the management interface.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.1(1).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug cdp events packets** command is issued:

```
switch# debug cdp events packets
Apr  8 21:22:34 cdp: Sent CDP packet, interface 0x2380000
Apr  8 21:22:34 cdp: Sent CDP packet, interface 0x2381000
Apr  8 21:22:35 cdp: Sent CDP packet, interface 0x2382000
Apr  8 21:22:35 cdp: Sent CDP packet, interface 0x2383000
Apr  8 21:22:51 cdp: Received CDP packet, interface 0x5000000
Apr  8 21:23:01 cdp: Sent CDP packet, interface 0x5000000
Apr  8 21:23:34 cdp: Sent CDP packet, interface 0x2380000
Apr  8 21:23:34 cdp: Sent CDP packet, interface 0x2381000
Apr  8 21:23:35 cdp: Sent CDP packet, interface 0x2382000
...
```

# debug core

To enable core demon debugging, use the **debug core** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug core error | flow

<b>Syntax Description</b>	<b>errors</b>	Enables debugging for core demon error conditions.
	<b>flow</b>	Enables debugging for the core demon flow.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug core flow** command is issued:

```
switch# debug core flow
```

# debug ethport

To enable Ethernet port debugging, use the **debug ethport** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```

debug ethport
  [ all ] |
  [ error ] |
  [ event (interface gigabitethernet slot-port) | module ( slot ) ] |
  [ ha (interface gigabitethernet slot-port) | module ( slot ) ] |
  [ trace (interface gigabitethernet slot-port) | module ( slot ) ]

```

Syntax Description		
<b>all</b>		Enables debugging for all Ethernet port features.
<b>error</b>		Enables debugging for Ethernet port error conditions.
<b>event</b>		Enables debugging for Ethernet port events.
<b>ha</b>		Enables debugging for port high availability.
<b>trace</b>		Enables debugging for Ethernet port traces.
<b>interface gigabitethernet</b> <i>slot-port</i>		Specifies the slot and port of the Gigabit Ethernet interface.
<b>module</b> <i>slot</i>		Specifies the slot number of the module being debugged.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug ethport all** command is issued:

```
switch# debug ethport all
```



# debug exceptionlog

To enable the exception log debugging feature, use the **debug exceptionlog** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug exceptionlog demux | deque | error | flow | info**

Syntax Description	
<b>demux</b>	Enables debugging for the exception logger demux functions.
<b>deque</b>	Enables debugging for the exception logger deque function.
<b>error</b>	Enables debugging for exception logger errors.
<b>flow</b>	Enables debugging for the exception logger flow.
<b>info</b>	Enables debugging for exception logger information.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug exceptionlog** command is issued:

```
switch# debug exceptionlog
7), credit(3), empty
```

# debug fc2

To enable debugging for the FC2 feature, use the **debug fc2** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```

debug fc2
  [ credit ] |
  [ error ( fcid fcid ) | ( interface fc type number | vsan vsan-id ) ] |
  [ flag ] |
  [ flow ( fcid fcid ) | ( interface fc type number | vsan vsan-id ) ] |
  [ frame ] |
  [ loopback ] |
  [ pkt ( both | tx | rx ) ( fcid fcid ) | ( interface fc type number | vsan vsan-id ) ] |
  [ pkthdr ( both | tx | rx ) ( fcid fcid ) | ( interface fc type number | vsan vsan-id ) ] |
  [ rdl ] |
  [ rxhdrhistory ( both | tx | rx ) ( fcid fcid ) | ( interface fc type number | vsan vsan-id ) ] |
  [ txhdrhistory ( both | tx | rx ) ( fcid fcid ) | ( interface fc type number | vsan vsan-id ) ]

```

Syntax	Description
<b>credit</b>	Enables FC2 credit debugging.
<b>error</b>	Enables FC2 error debugging.
<b>flag</b>	Enables FC2 flags debugging.
<b>flow</b>	Enables FC2 flow debugging.
<b>frame</b>	Enables FC2 frame debugging.
<b>loopback</b>	Enables FC2 loopback debugging.
<b>pkt</b>	Enables FC packet debugging.
<b>pkthdr</b>	Enables FC header debugging.
<b>rdl</b>	Enables FC2 RDL debugging.
<b>rxhdrhistory</b>	Enables FC2 received header history debugging.
<b>txhdrhistory</b>	Enables FC2 transmitted header history debugging.
<b>both</b>	Enables debugging in both the transmit and receive directions.
<b>tx</b>	Enables debugging in the transmit direction.
<b>rx</b>	Enables debugging in the receive direction.
<b>fcid</b> <i>fcid</i>	Restricts debugging to the specified FCID.
<b>interface fc</b> <i>type number</i>	Restricts debugging to the specified interface.
<b>vsan</b> <i>vsan-id</i>	Restricts debugging to the specified VSAN.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** If FSPF receives a bad FC2 packet analyze the output of the **debug fc2 pkt** command.

**Examples** The following example displays the system output when the **debug fc2 error vsan 1** command is issued:

```
switch1# debug fc2 error vsan 1
```

# debug fcc

To enable debugging for the Fibre Channel Congestion (FCC) function, use the **debug fcc** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug fcc
  [ all ] |
  [ errors ( module slot ) ] |
  [ events ( module slot ) ] |
  [ mts { pkt both | tx | rx ( node range | opcode range | sap range ) } | { ( pkthdr both | tx | rx
  ( numpkt range ) } ]
  [ trace ( module slot ) ]]
```

Syntax	Description
<b>all</b>	Enables debugging for all FCC features.
<b>errors</b>	Enables debugging for FCC error conditions.
<b>events</b>	Enables debugging for FCC events.
<b>mts</b>	Enables debugging for FCC tx/rx MTS packets.
<b>trace</b>	Enables debugging for FCC traces.
<b>module slot</b>	Specifies the slot number of the module being debugged.
<b>pkt</b>	Enables debugging for FCC tx/rx FCC packets.
<b>pkthdr</b>	Enables debugging for FCC tx/rx FCC headers.
<b>numpkt</b>	Specifies the number of required packets
<b>both</b>	Specifies debugging in both the transmit and receive directions.
<b>tx</b>	Specifies debugging in the transmit direction,
<b>rx</b>	Specifies debugging in the receive direction.
<b>node</b>	Specifies the node for the packets in the receive direction.
<b>opcode</b>	Specifies the opcode for the packets in the receive direction.
<b>sap</b>	Specifies the sap for the packets in the receive direction.
<b>range</b>	Specifies the integer range from 1 to 4095.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples**

The following example displays the system output when the **debug fcc all** command is issued:

```
switch# debug fcc all
```

# debug fcdomain

To enable debugging for the fcdomain feature, use the **debug fcdomain** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug fcdomain

```
[all] |
[critical] |
[error]
[fc (pkt | pkthdr) (both | tx | rx) (interface type number | vsan vsan-id) ] |
[ipc (pkt | pkthdr) (both | tx | rx) (interface type number | vsan vsan-id) ] |
[memory] |
[notify] |
[phase]
```

Syntax	Description
<b>all</b>	Enables debugging of all fcdomain parameters.
<b>critical</b>	Enables debugging of critical operations.
<b>error</b>	Enables debugging of error operation.
<b>fc</b>	Enables debugging of Fibre Channel Packets and Headers.
<b>ipc</b>	Enables debugging of IPC Packets and Headers.
<b>pkt</b>	Enables debugging of packets.
<b>pkthdr</b>	Enables debugging of headers.
<b>both</b>	Enables debugging in both the transmit and receive directions.
<b>tx</b>	Enables debugging in the transmit direction,
<b>rx</b>	Enables debugging in the receive direction.
<b>interface</b> <i>type number</i>	Specifies the interface to be debugged.
<b>vsan</b> <i>vsan-id</i>	Restricts debugging to the specified VSAN.
<b>memory</b>	Enables debugging of memory operations.
<b>notify</b>	Enables debugging of notifications
<b>phase</b>	Enables debugging of global phases

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples**

The following example displays the system output when the **debug fcdomain critical** command is issued:

```
switch# debug fcdomain critical
Jan 27 07:04:31 fcdomain: Src: 0x00000501/6243 Dst: 0x00000501/14 ID: 0x0005BF
41 Size: 252 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x0005BF41 HA_SEQNO:
0x00000000 TS: 0x183C4D027F4A3
Jan 27 07:04:31 fcdomain: 00 00 00 00 68 00 00 00 00 00 00 00 00 00 00 00
Jan 27 07:04:31 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Jan 27 07:04:31 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
Jan 27 07:04:31 fcdomain: 2F 64 65 76 2F 70 74 73 2F 30 00 00 00 00 00 00
Jan 27 07:04:31 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
...
```

The following example displays the system output when the **debug fcdomain error** command is issued:

```
switch# debug fcdomain error
Jan 27 07:05:29 fcdomain: Src: 0x00000501/6245 Dst: 0x00000501/14 ID: 0x0005BF
7E Size: 252 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x0005BF7E HA_SEQNO:
0x00000000 TS: 0x183D5E63C081A
Jan 27 07:05:29 fcdomain: 00 00 00 00 64 00 00 00 00 00 00 00 00 00 00 00
Jan 27 07:05:29 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Jan 27 07:05:29 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
Jan 27 07:05:29 fcdomain: 2F 64 65 76 2F 70 74 73 2F 30 00 00 00 00 00 00
...
```

The following example displays the system output when the **debug fcdomain ipc pkthdr both** command is issued:

```
vegas2# debug fcdomain ipc pkthdr both
Apr 8 20:44:38 fcdomain: Src: 0x00000501/3883 Dst: 0x00000501/14 ID: 0x00038E
1D Size: 252 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x00038E1D HA_SEQNO:
0x00000000 TS: 0x5DD9B14EA3AA REJ:0
Apr 8 20:44:38 fcdomain: 00 00 00 00 08 00 00 00 00 00 00 00 00 00 00 00
Apr 8 20:44:38 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 8 20:44:38 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
Apr 8 20:44:38 fcdomain: 2F 64 65 76 2F 70 74 73 2F 30 00 00 00 00 00 00
Apr 8 20:44:38 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 8 20:44:38 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
...
Apr 8 20:44:38 fcdomain: Src: 0x00000501/3883 Dst: 0x00000501/14 ID: 0x00038E
20 Size: 252 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x00038E20 HA_SEQNO:
0x00000000 TS: 0x5DD9B186CCEB REJ:0
Apr 8 20:44:38 fcdomain: 00 00 00 00 07 00 00 00 00 00 00 00 00 00 00 00
Apr 8 20:44:38 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 8 20:44:38 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
Apr 8 20:44:38 fcdomain: 2F 64 65 76 2F 70 74 73 2F 30 00 00 00 00 00 00
Apr 8 20:44:38 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 8 20:44:38 fcdomain: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
...
```

**Related Commands**

Command	Description
<b>show fcdomain domain-list</b>	Displays current domains in the fabric.
<b>fcdomain</b>	Enables fcdomain features.

# debug fcfwd

To enable debugging for the Fibre Channel forwarding feature, use the **debug fcfwd** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug fcfwd

```
[ flogimap error ( module slot | vsan vsan-id ) | event ( module slot | vsan vsan-id ) | trace (
module slot | vsan vsan-id ) ] |
[ idxmap error ( module slot | vsan vsan-id ) | event ( module slot | vsan vsan-id ) | trace (
module slot | vsan vsan-id ) ] |
[ pemap error ( module slot | vsan vsan-id ) | event ( module slot | vsan vsan-id ) | trace (
module slot | vsan vsan-id ) ] |
[ sfib error ( module slot | vsan vsan-id ) | event ( module slot | vsan vsan-id ) | trace ( module
slot | vsan vsan-id ) ] |
[ spanmap error ( module slot | vsan vsan-id ) | event ( module slot | vsan vsan-id ) | trace (
module slot | vsan vsan-id ) ]
```

Syntax	Description
<b>flogimap</b>	Enables flogimap debugging.
<b>idxmap</b>	Enables idxmap debugging.
<b>pemap</b>	Enables pemap debugging.
<b>sfib</b>	Enables sfib debugging.
<b>spanmap</b>	Enables spanmap debugging.
<b>error</b>	Enables debugging for FCC error conditions.
<b>event</b>	Enables debugging for FCC events.
<b>trace</b>	Enables debugging for FCC traces.
<b>module slot</b>	Specifies the slot number of the module being debugged.
<b>vsan vsan-id</b>	Restricts debugging to the specified VSAN.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug fcfwd error** command is issued:

```
switch# debug fcfwd error
```



# debug fcns

To enable debugging for name server registration, use the **debug fcns** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug fcns
  [ all (vsan vsan-id) ] |
  [ errors (vsan vsan-id) ] |
  [ events mts (vsan vsan-id) | query (vsan vsan-id) | register (vsan vsan-id) ]
```

Syntax Description	
<b>all</b>	Enables debugging for all name server features.
<b>errors</b>	Enables debugging for name server error conditions.
<b>events</b>	Enables debugging for name server events.
<b>mts</b>	Enables debugging for name server tx/rx MTS packets.
<b>query</b>	Enables debugging for name server tx/rx CDP packets.
<b>register</b>	Enables debugging for name server PSS related events.
<b>vsan vsan-id</b>	Restricts debugging to the specified VSAN.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug fcns events register vsan 99** command is issued:

```
switch# debug fcns events register vsan 99
Feb 17 04:42:54 fcns: vsan 99: Got Entry for port-id 27800
Feb 17 04:42:54 fcns: vsan 99: Registered port-name 36a4078be0000021 for port-id 780200
Feb 17 04:42:54 fcns: vsan 99: Registered node-name 36a4078be0000020 for port-id 780200
...
```

# debug fcs

To enable debugging for the fabric configuration server, use the **debug fcs** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```

debug fcs
  [ all (vsan vsan-id) ] |
  [ discovery events ] |
  [ errors (vsan vsan-id) ] |
  [ mts events (brief | detail) ] |
  [ pss events ] |
  [ queries events (vsan vsan-id) ] |
  [ registration events (vsan vsan-id) ] |
  [ rscn events (vsan vsan-id) ] |
  [ snmp events ]

```

Syntax	Description
<b>all</b>	Enables debugging for all FCS features.
<b>discovery events</b>	Enables debugging for FCS discovery events.
<b>errors</b>	Enables debugging for FCS error conditions.
<b>mts events</b>	Enables debugging for FCS tx/rx MTS events.
<b>brief</b>	Provides brief information for each event.
<b>detail</b>	Provides detailed information for each event.
<b>queries events</b>	Enables debugging for FCS tx/rx events.
<b>registration events</b>	Enables debugging for FCS PSS related events.
<b>rscn events</b>	Enables debugging for FCS RSCN events.
<b>snmp events</b>	Enables debugging for FCS SNMP events.
<b>vsan vsan-id</b>	Restricts debugging to the specified VSAN.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug fcs all** command is issued:

```
switch# debug fcs all
```

# debug flogi

To enable debugging for the fabric login (FLOGI) feature, use the **debug flogi** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```

debug flogi
  [ action (interface type number | vsan vsan-id ) ]
  [ all ] |
  [ demux (interface type number | vsan vsan-id ) ]
  [ error ] |
  [ event (interface type number | vsan vsan-id ) ] |
  [ ha (interface type number | vsan vsan-id ) ] |
  [ init (interface type number | vsan vsan-id ) ] |
  [ timers (interface type number | vsan vsan-id ) ]
  [ trace (interface type number | vsan vsan-id) ]
  [ warning ]

```

Syntax Description		
<b>action</b>		Enables all FLOGI debug features.
<b>all</b>		Enables all FLOGI debug options.
<b>demux</b>		Enables FLOGI demux
<b>error</b>		Enables debugging for FLOGI error conditions.
<b>event</b>		Enables debugging for FLOGI FSMs and events.
<b>ha</b>		Enables debugging for FLOGI high availability.
<b>init</b>		Enables debugging of FLOGI addition, deletion, and initialization.
<b>timer</b>		Enables debugging for FLOGI message timers
<b>trace</b>		Enables debugging for FLOGI traces.
<b>warning</b>		Enables debugging for FLOGI warnings.
<b>interface</b> <i>type number</i>		Restricts debugging to the specified interface.
<b>vsan</b> <i>vsan-id</i>		Restricts debugging to the specified VSAN.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

---

**Examples**

The following example displays the system output when the **debug flogi all** command is issued:

```
switch# debug flogi all
Apr  9 22:44:08 flogi: fs_demux: msg consumed by sdwrap_process msg
Apr  9 22:44:08 flogi: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
Apr  9 22:44:08 flogi: fu_fsm_execute_all: null fsm_event_list
Apr  9 22:44:08 flogi: fu_fsm_engine: mts msg MTS_OPC_DEBUG_WRAP_MSG(msg_id 67690) dropped
```

The following example displays the system output when the **debug flogi event** command is issued:

```
switch# debug flogi event
Apr 10 00:07:16 flogi: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
Apr 10 00:07:16 flogi: fu_fsm_execute_all: null fsm_event_list
Apr 10 00:07:16 flogi: fu_fsm_engine: mts msg MTS_OPC_DEBUG_WRAP_MSG(msg_id 71314) dropped
```

The following example displays the system output when the **debug flogi trace** command is issued:

```
switch# debug flogi trace
Apr 10 00:42:36 flogi: fs_genport_vsan_hash_fn: key: 0x1 index: 0x1
Apr 10 00:42:36 flogi: fs_mts_hdlr_fs_flogo: FLOGI HOLD(0x8122144) refcnt:3
Apr 10 00:42:36 flogi: fs_clear_all_outstanding_responses_for_flogi: FLOGI FREE(
a07e00300500252b) refcnt:3
```

# debug fspf

To enable debugging for the FSPF feature, use the **debug fspf** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug fspf

```
[ all (interface type number | vsan vsan-id ) ] |
[ database ] |
[ error (interface type number | vsan vsan-id ) ] |
[ event (interface type number | vsan vsan-id ) ] |
[ fc (pkt | pkthdr) (both | tx | rx) (interface type number | vsan vsan-id) ] |
[ flood (interface type number | vsan vsan-id) ] |
[ ha (interface type number | vsan vsan-id) ] |
[ mts (pkt | pkthdr) (both | tx | rx) (interface type number | vsan vsan-id) ] |
[ retrans (interface type number | vsan vsan-id) ] |
[ route ] |
[ timer ]
```

Syntax Description		
<b>all</b>		Enables debugging for all FSPF features.
<b>database</b>		Enables debugging for the FSPF database.
<b>error</b>		Enables debugging for FSPF error conditions.
<b>events</b>		Enables debugging for FSPF events.
<b>fc</b>		Enables debugging of Fibre Channel Packets and Headers.
<b>ipc</b>		Enables debugging of IPC Packets and Headers.
<b>pkt</b>		Enables debugging of packets.
<b>pkthdr</b>		Enables debugging of headers.
<b>both</b>		Enables debugging in both the transmit and receive directions.
<b>tx</b>		Enables debugging in the transmit direction.
<b>rx</b>		Enables debugging in the receive direction.
<b>flood</b>		Enables debugging for FSPF flooding events.
<b>ha</b>		Enables debugging for FSPF high availability.
<b>mts</b>		Enables debugging for FSPF tx/rx MTS events.
<b>retrans</b>		Enables debugging for FSPF retransmits.
<b>route</b>		Enables debugging for FSPF route computation.
<b>timer</b>		Enables debugging for FSPF timers.
<b>interface</b> <i>type number</i>		Restricts debugging to the specified interface.
<b>vsan</b> <i>vsan-id</i>		Restricts debugging to the specified VSAN.

**Defaults** None.

**Command Modes** EXEC mode.

---

**Command History**

This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

---

**Usage Guidelines**

If you receive bad packets on an interface, use the **debug fc pkt** command.

If you receive an error in processing a packet on an interface in VSAN, turn on **debug fspf error** to get more information. Make sure there is no misconfiguration of FSPF parameters on the two ends of the interface. Also issue the **debug fspf fc pkt** command for the specific interface.

If you receive an error in flooding the local LSR in a VSAN issue the **debug fspf flood** and **debug fspf error** commands. If error is reported in transmitting packet check if interface is up and turn on **debug fc2 error**.

If you receive an error in processing a timer event for the interface in a VSAN, issue the **debug fspf error** command.

If you receive an error in processing due to a wrong MTS message, use the **debug fspf mts pkt** and **debug fspf error** commands.

If you receive an error when interacting with RIB, use the **debug fspf route** command along with the RIB debug traces.

If you receive an error in computing routes for VSANs, issue the **debug fspf error** and the **debug fspf route** commands.

If you receive an error due to the interface being stuck in a state other than FULL, use the **debug fspf event** and **debug fspf fc pkt** commands on the interfaces involved.

---

**Examples**

The following example displays the system output when the **debug fspf all** command is issued:

```
switch1# debug fspf all
Apr 5 11:50:01 fspf: Wrong hello interval for packet on interface 100f000 in VSAN 1
Apr 5 11:50:04 fspf: Error in processing hello packet , error code = 4
```

# debug hardware

To configure debugging for the hardware kernel module parameters, use the **debug hardware** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug hardware
  [ arbiter error | flow ] |
  [ sso flow | init | interrupt ]
```

Syntax Description		
<b>arbiter</b>		Enables debugging for the hardware arbiter driver.
<b>sso</b>		Enables debugging for the SSO driver.
<b>errors</b>		Enables debugging for hardware kernel errors.
<b>flow</b>		Enables debugging for hardware flow errors
<b>init</b>		Enables debugging for hardware initialization.
<b>interrupt</b>		Enables debugging for hardware interrupts.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug hardware arbiter error group** command is issued:

```
switch# debug hardware arbiter error group 1
```

# debug idehsd

To enable IDE hot swap handler debugging, use the **debug idehsd** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug idehsd
  [ cmd dbglevel debug-level ] |
  [ error ] |
  [ flow ]
```

## Syntax Description

<b>cmd</b>	Enables debugging for the IDE hot swap handler.
<b>dbglevel</b> <i>debug-level</i>	Specifies the debug level (0 to 8).
<b>error</b>	Enables debugging for IDE hot swap handler error conditions.
<b>flow</b>	Enables debugging for IDE hot swap handler flow.

## Defaults

None.

## Command Modes

EXEC mode.

## Command History

This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

## Usage Guidelines

None.

## Examples

The following example displays the system output when the **debug idehsd cmd dbglevel** command is issued:

```
switch# debug idehsd cmd dbglevel 5
set debug level to 5 succeeded
```



# debug ipconf

To enable IP configuration debugging, use the **debug ipconf** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug ipconf all | errors | events | info | pss**

Syntax Description		
<b>all</b>		Enables debugging for all IP configuration features.
<b>errors</b>		Enables debugging for IP configuration error conditions.
<b>events</b>		Enables debugging for IP configuration tx/rx MTS events.
<b>info</b>		Enables debugging for IP configuration information.
<b>pss</b>		Enables debugging for IP configuration PSS operations.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug ipconf all** command is issued:

```
switch# debug ipconf all
```

# debug ipfc

To enable IPFC debugging, use the **debug ipfc** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug ipfc
  [ all ] |
  [ errors ] |
  [ events ] |
  [ info ] |
  [ kernel (errors | events) ]
```

## Syntax Description

<b>all</b>	Enables debugging for all IPFC features.
<b>errors</b>	Enables debugging for IPFC error conditions.
<b>events</b>	Enables debugging for IPFC tx/rx MTS events.
<b>info</b>	Enables debugging for IPFC n information.
<b>kernel</b>	Enables debugging for IPFC PSS operations.
<b>errors</b>	Enables debugging for IPFC kernel error conditions.
<b>events</b>	Enables debugging for IPFC kernel events.

## Defaults

None.

## Command Modes

EXEC mode.

## Command History

This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

## Usage Guidelines

None.

## Examples

The following example displays the system output when the **debug ipfc kernel errors** command is issued:

```
switch# debug ipfc kernel errors
```

# debug ips

To enable debugging for the IP storage (IPS) manager, use the **debug ips** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug ips
  [ all ] |
  [ demux ]
  [ error ] |
  [ flow ( ethernet | fcip | iscsi | iscsi_detail ) ] |
  [ fsm ] |
  [ ha ] |
  [ init ] |
  [ show_all ] |
  [ timers ]
  [ warning ]
```

Syntax	Description
<b>all</b>	Enables all IPS debug options.
<b>demux</b>	Enables IPS demux
<b>error</b>	Enables debugging for IPS error conditions.
<b>flow</b>	Enables debugging for the IPS flow.
<b>ethernet</b>	Restricts debugging to the Ethernet flow
<b>fcip</b>	Restricts debugging to the FCIP flow
<b>iscsi</b>	Restricts debugging to the iSCSI flow
<b>iscsi_detail</b>	Restricts debugging to a detailed iSCSI flow
<b>fsm</b>	Enables debugging for IPS FSM and events.
<b>ha</b>	Enables debugging for IPS high availability.
<b>init</b>	Enables debugging of IPS addition, deletion, and initialization.
<b>show_all</b>	Enables all debugging IPS manager flags.
<b>timers</b>	Enables debugging for FLOGI message timers
<b>warning</b>	Enables debugging for FLOGI warnings.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.1(1).

**Usage Guidelines** None.

---

**Examples**

The following example displays the system output when the **debug ips show\_all** command is issued:

```
switch# debug ips show_all
IPS Manager:
iSCSI Trace Detail debugging is on
```

# debug logfile

To direct the output of the debug commands to a specified file, use the **debug logfile** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug logfile** *filename* ( *size size* )

<b>Syntax Description</b>	<i>filename</i>	Assigns the name of the log file.
	<b>size size</b>	Specifies the logfile size in bytes (4096-4194304).

**Defaults** Disabled.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** Use this command to log debug messages to a special log file. This file is more secure and easier to process than sending the debug output to the console.

**Examples** The following example redirects the output of the debug commands to the file named *sample*.

```
switch# debug logfile sample
```

The following example assigns the log file size for the file named *sample*.

```
switch# debug logfile sample size 410000
```

# debug mcast

To enable debugging for multicast definitions, use the **debug mcast** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug mcast
  [ all ] |
  [ error (vsan vsan-id) (interface fc slot-port) ] |
  [ event (vsan vsan-id) (interface fc slot-port) ] |
  [ mts { pkt both | tx | rx } ( node range | opcode range | sap range ) ] { ( pkthdr both | tx |
  rx ( numpkt range ) ) } |
  [ trace (vsan vsan-id) (interface fc slot-port) ]
```

Syntax	Description
<b>all</b>	Enables debugging for all multicast definitions.
<b>error</b>	Enables debugging for multicast errors.
<b>event</b>	Enables debugging for multicast events.
<b>mts</b>	Enables debugging for multicast tx/rx MTS events.
<b>trace</b>	Enables debugging for multicast traces.
<b>vsan vsan-id</b>	Restricts debugging to the specified VSAN.
<b>interface fc slot-port</b>	Restricts debugging to the specified interface.
<b>pkt</b>	Specifies debugging of packets.
<b>pkthdr</b>	Specifies debugging of headers.
<b>numpkt</b>	Specifies the number of required packets
<b>both</b>	Specifies debugging in both the transmit and receive directions.
<b>tx</b>	Specifies debugging in the transmit direction,
<b>rx</b>	Specifies debugging in the receive direction.
<b>node</b>	Specifies the node for the packets in the receive direction.
<b>opcode</b>	Specifies the opcode for the packets in the receive direction.
<b>sap</b>	Specifies the sap for the packets in the receive direction.
<i>range</i>	Specifies the integer range from 1 to 4095.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples**

The following example displays the system output when the **debug mcast all** command is issued:

```
switch# debug mcast all
```

# debug mip

To enable debugging for multiple IP (MIP) kernel drivers, use the **debug mip** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug mip errors | events

Syntax Description	
<b>errors</b>	Enables debugging for MIP error conditions.
<b>events</b>	Enables debugging for MIP events.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug mip errors** command is issued:

```
switch# debug mip errors
```



# debug module

To enable debugging for switching or service modules, use the **debug module** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug module
  [ all ] |
  [ error ( module slot ) ] |
  [ event ] |
  [ ha ] |
  [ no-powerdown ] |
  [ trace ( module slot ) ]
```

Syntax Description		
<b>all</b>		Enables debugging for all module features.
<b>error</b>		Enables debugging for module error conditions.
<b>event</b>		Enables debugging for module events.
<b>ha</b>		Enables debugging for a module's high availability features.
<b>no-powerdown</b>		Disables the power cycle feature for the module.
<b>trace</b>		Enables debugging for a module's trace flows.
<b>module slot</b>		Restricts debugging to the specified module.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug fcs all** command is issued:

```
switch# debug fcs all
Apr 28 19:23:20 module: fu_fsm_execute: (ID(5): Slot 4, node 0x0402)
Apr 28 19:23:20 module:      current state [LCM_ST_LC_ONLINE]
Apr 28 19:23:20 module:      current event [LCM_EV_LCM_HEARTBEAT_TIMEOUT]
Apr 28 19:23:20 module:      next state   [LCM_ST_LC_ONLINE]
Apr 28 19:23:20 module: fu_add_pss_data: adding data for key <8, 0x33000000000000
004> to the pss runtime service add data list
Apr 28 19:23:20 module: fu_add_pss_data: added key <8, 0x33000000000000004> data
...
```

# debug ntp

To enable debugging for the NTP module, use the **debug ntp** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug ntp errors | info

Syntax Description	errors	info
	Enables debugging for NTP error conditions.	Enables debugging for NTP information and events.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug ntp errors** command is issued:

```
switch# debug ntp errors
```

# debug platform

To enable debugging for the platform manager, use the **debug platform** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug platform
  [ all (fc_id fc-id) ] |
  [ error (module slot) ] |
  [ flow (module slot) ]
  [ fsm ] |
  [ ha ] |
  [ hitless ] |
  [ mts (pkt | pkthdr) (both | tx | rx) ]
```

Syntax	Description
<b>all</b>	Enables debugging for all platform features.
<b>error</b>	Enables debugging for platform-related error conditions.
<b>flow</b>	Enables debugging for platform-related flows.
<b>fsm</b>	Enables debugging for platform-related FSMs.
<b>ha</b>	Enables debugging for platform-related high availability.
<b>hitless</b>	Enables the platform loading feature while the switch is in hitless mode.
<b>mts</b>	Enables debugging for platform-related tx/rx MTS events.
<b>fcid</b> <i>fc-id</i>	Restricts debugging to the specified FC ID module number (from 0 to 2147483647).
<b>pkt</b>	Enables debugging of packets.
<b>pkthdr</b>	Enables debugging of headers.
<b>both</b>	Enables debugging in both the transmit and receive directions.
<b>tx</b>	Enables debugging in the transmit direction.
<b>rx</b>	Enables debugging in the receive direction.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug platform all** command is issued:

```
switch# debug platform all
```

# debug port

To enable debugging for ports, use the **debug port** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```

debug port
  [ all ] |
  [ error ] |
  [ event (interface type number | vsan vsan-id ) ] |
  [ ha (interface type number | vsan vsan-id ) ] |
  [ trace (interface type number | vsan vsan-id)]

```

Syntax Description	action	Enables all port debug features.
	<b>all</b>	Enables all port debug options.
	<b>error</b>	Enables debugging for port error conditions.
	<b>event</b>	Enables debugging for port FSMs and events.
	<b>ha</b>	Enables debugging for port high availability.
	<b>trace</b>	Enables debugging for port traces.
	<b>interface</b> <i>type number</i>	Restricts debugging to the specified interface.
	<b>vsan</b> <i>vsan-id</i>	Restricts debugging to the specified VSAN.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

---

**Examples**

The following example displays the system output when the **debug port all** command is issued:

```
switch# debug port all
Apr 10 00:49:38 port: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
Apr 10 00:49:38 port: fu_fsm_execute_all: null fsm_event_list
Apr 10 00:49:38 port: fu_fsm_engine: mts msg MTS_OPC_DEBUG_WRAP_MSG(msg_id 40239) dropped
```

The following example displays the system output when the **debug port event** command is issued:

```
switch# debug port event
Apr 10 15:30:35 port: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
Apr 10 15:30:35 port: fu_fsm_execute_all: null fsm_event_list
Apr 10 15:30:35 port: fu_fsm_engine: mts msg MTS_OPC_DEBUG_WRAP_MSG(msg_id 7002)
dropped
switch# Apr 10 15:30:35 port: fu_priority_select: - setting fd[3] for select call -
setting fd[5] for select call - setting fd[6] for select call
Apr 10 15:30:35 port: fu_priority_select_select_queue: round credit(16)
Apr 10 15:30:35 port: curr_q - FU_PSEL_Q_CAT_FD, usr_q_info(32), fd(5), priority(3),
credit(2), empty
Apr 10 15:30:35 port: fu_priority_select: returning FU_PSEL_Q_CAT_MTS queue, fd(3),
usr_q_info(8)
```

# debug port-channel

To enable debugging for PortChannels, use the **debug port-channel** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug port-channel
  [ all (interface port-channel number) ] |
  [ error ] |
  [ event (interface port-channel number) ] |
  [ ha (interface port-channel number) ] |
  [ mts (pkt | pkthdr) (both | tx | rx) (interface port-channel number) ]
  [ query ]
  [ trace (interface port-channel number) ]
```

Syntax Description		
<b>all</b>		Enables all PortChannel debug options.
<b>error</b>		Enables debugging for PortChannel error conditions.
<b>event</b>		Enables debugging for PortChannel FSMs and events.
<b>ha</b>		Enables debugging for PortChannel high availability.
<b>pkt</b>		Enables debugging of packets.
<b>pkthdr</b>		Enables debugging of headers.
<b>both</b>		Enables debugging in both the transmit and receive directions.
<b>tx</b>		Enables debugging in the transmit direction.
<b>rx</b>		Enables debugging in the receive direction.
<b>trace</b>		Enables debugging for PortChannel traces.
<b>interface port-channel number</b>		Restricts debugging to the specified PortChannel.
<b>vsan vsan-id</b>		Restricts debugging to the specified VSAN.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug port-channel all** command is issued:

```
switch# debug port-channel all
```

# debug qos

To enable debugging for quality of Service (QoS), use the **debug qos** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug qos all | detail | errors | flow | trace**

Syntax Description	all	Enables all QoS debug options.
	<b>detail</b>	Enables all QoS debug output.
	<b>error</b>	Enables debugging for QoS error conditions.
	<b>flow</b>	Enables flow-level QoS debug options.
	<b>trace</b>	Enables debugging for QoS traces.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug qos all** command is issued:

```
switch# debug qos all
```

## debug rd-reg

To enable debugging for the list of devices using the read-register feature, use the **debug rd-reg** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug rd-reg** ( *device-name* | *register address* )

Syntax Description		
	<i>device-name</i>	Specifies the device name for the required device.
	<i>register address</i>	Specifies the register address for the required device.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug rd-reg abc** command is issued:

```
switch# debug rd-reg abc
```



# debug rdl

To enable debugging for the list of devices using the read-register feature, use the **debug rdl** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug rdl errors

<b>Syntax Description</b>	<b>errors</b> Enables debugging for RDL errors.
<b>Defaults</b>	None.
<b>Command Modes</b>	EXEC mode.
<b>Command History</b>	This command was introduced in Cisco MDS SAN-OS Release 1.0(2).
<b>Usage Guidelines</b>	None.
<b>Examples</b>	The following example displays the system output when the <b>debug rdl errors</b> command is issued: <pre>switch# <b>debug rdl errors</b></pre>

# debug rib

To enable debugging for the routing information base (RIB) feature, use the **debug rib** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug rib all | error | event | trace**

Syntax Description		
<b>all</b>		Enables debugging for all RIB features.
<b>error</b>		Enables debugging for RIB errors.
<b>event</b>		Enables debugging for RIB events.
<b>trace</b>		Enables debugging for trace events.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** If a RIB operation is ignored or not supported, then issue the **debug rib all** command to find out more details.

**Examples** The following example displays the system output when the **debug rib error** command is issued:

```
switch# debug rib error
```

# debug rscn

To enable debugging for the registered state change notification (RSCN) feature, use the **debug rscn** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug rscn
  [ all (vsan vsan-id) ] |
  [ errors (vsan vsan-id) ] |
  [ events (vsan vsan-id) ] |
  [ mts-errors (vsan vsan-id) ] |
  [ mts-events (vsan vsan-id) ]
```

Syntax Description		
<b>all</b>		Enables debugging for all RSCN features.
<b>error</b>		Enables debugging for RSCN errors.
<b>event</b>		Enables debugging for RSCN events.
<b>mts-errors</b>		Enables debugging for RSCN MTS errors.
<b>mts-events</b>		Enables debugging for RSCN MTS events.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug rscn error** command is issued:

```
switch# debug rscn error
```

# debug scsi-target

To enable debugging for SCSI targets, use the **debug scsi-target** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug scsi-target error | flow

Syntax Description	error	flow
	Enables debugging for SCSI target daemon errors.	Enables debugging for the SCSI target flow.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.1(1).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug scsi-target flow** command is issued:

```
switch# debug scsi-target flow
Apr 28 21:11:52 vhbaid: vhba_mts_handler: sdwrap_dispatch: retval:0
Apr 28 21:11:54 vhbaid: vhbaid_handle_timeout: timer:1 context:(nil)
Apr 28 21:12:06 vhbaid: vhba_mts_handler: sysmgr_dispatch: retval:-1
```

# debug security

To enable debugging for the security and accounting features, use the **debug security** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug security all | events | mts | radius**

Syntax Description		
	<b>all</b>	Enables debugging for all security features.
	<b>events</b>	Enables debugging for security events.
	<b>mts</b>	Enables debugging for security MTS packets.
	<b>radius</b>	Enables debugging for RADIUS events.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug security radius** command is issued:

```
switch# debug security radius
Mar  5 00:51:13 securityd: RADIUS is enabled, hence it will be tried first for CHAP
authentication
Mar  5 00:51:13 securityd: reading RADIUS configuration
Mar  5 00:51:13 securityd: opening radius configuration for group:default
Mar  5 00:51:13 securityd: opened the configuration successfully
Mar  5 00:51:13 securityd: GET request for RADIUS global config
Mar  5 00:51:13 securityd: got back the return value of global radius configuration
operation:success
Mar  5 00:51:13 securityd: closing RADIUS pss configuration
Mar  5 00:51:13 securityd: opening radius configuration for group:default
```

# debug sensor

To enable debugging for the sensor manager, use the **debug sensor** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug sensor demux | deque | error | info | init**

Syntax Description		
<b>demux</b>		Enables debugging for sensor demux functions.
<b>deque</b>		Enables debugging for sensor deque events.
<b>error</b>		Enables debugging for sensor errors.
<b>info</b>		Enables debugging for sensor information.
<b>init</b>		Enables debugging for sensor initialization.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** Use this command to debug sensor manager events and information.

**Examples** The following example displays the system output when the **debug sensor info** command is issued:

```
switch# debug sensor info
```

Related Commands	Command	Description
	<b>show environment temperature</b>	Displays current temperature threshold settings and state.

# debug snmp

To enable debugging for the SNMP manager, use the **debug snmp** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug snmp
  [ all ] |
  [ errors ] |
  [ mts ( pkt both | tx | rx ( node range | opcode range | sap range ) | { pkthdr both | tx | rx (
  numpkt range ) } ] |
  [ pkt-dump ] |
  [ trace ( trace-entryexit | trace-stub ) ]
```

Syntax	Description
<b>all</b>	Enables debugging for all SNMP output.
<b>errors</b>	Enables debugging for SNMP error output.
<b>mts</b>	Enables debugging for SNMP packets and headers.
<b>pkt-dump</b>	Enables a packet dump of debug output.
<b>trace</b>	Enables trace level debug output.
<b>pkt</b>	Specifies debugging of packets.
<b>pkthdr</b>	Specifies debugging of headers.
<b>both</b>	Specifies debugging in both the transmit and receive directions.
<b>tx</b>	Specifies debugging in the transmit direction.
<b>rx</b>	Specifies debugging in the receive direction.
<b>node</b>	Specifies the node for the packets in the receive direction.
<b>opcode</b>	Specifies the opcode for the packets in the receive direction.
<b>sap</b>	Specifies the sap for the packets in the receive direction.
<b>numpkt</b>	Specifies the number of required packets
<i>range</i>	Specifies the integer range from 1 to 4095.
<b>trace-entryexit</b>	Specifies trace-level entry or exit debug output.
<b>trace-stub</b>	Specifies trace-level stub debug output.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

---

**Examples**

The following example displays the system output when the **debug snmp trace** command is issued:

```
switch# debug snmp trace  
Apr 29 16:03:34 snmpd[1177]: SDWRAP message Successfully processed
```



# debug span

To enable SPAN debugging, use the **debug span** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug span all | error | event | trace | warning**

Syntax Description		
	<b>all</b>	Enables debugging for all SPAN features.
	<b>error</b>	Enables debugging for SPAN errors.
	<b>event</b>	Enables debugging for SPAN events.
	<b>trace</b>	Enables debugging for SPAN traces.
	<b>warning</b>	Enables debugging for SPAN warning messages.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug span all** command is issued:

```
switch# debug span all
Apr 29 16:06:44 span: span_demux: msg consumed by sdwrap_process msg
Apr 29 16:06:44 span: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
Apr 29 16:06:44 span: fu_fsm_execute_all: null fsm_event_list
Apr 29 16:06:44 span: fu_fsm_engine: mts msg MTS_OPC_DEBUG_WRAP_MSG(msg_id 2548887)
dropped
Apr 29 16:06:48 span: fu_priority_select: - setting fd[3] for select call
Apr 29 16:06:48 span: fu_priority_select_select_queue: round credit(12)
Apr 29 16:06:48 span: curr_q - FU_PSEL_Q_CAT_CQ, usr_q_info(4), priority(7),
credit(6), empty
Apr 29 16:06:48 span: fu_priority_select: returning FU_PSEL_Q_CAT_MTS queue, fd(3),
usr_q_info(2)
Apr 29 16:06:48 span: span_get_data_from_mts_q dequeued mts msg (26e525),
MTS_OPC_DEBUG_WRAP_MSG
```

# debug system

To enable system debugging, use the **debug system** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug system [ all | error | ha | health ( all | error | event | ha | trace ) ]
```

Syntax Description		
<b>all</b>		Enables all PortChannel debug options.
<b>error</b>		Enables debugging for PortChannel error conditions.
<b>ha</b>		Enables debugging for PortChannel high availability.
<b>health</b>		Enables online health monitoring debugging.
<b>all</b>		Specifies debugging of all health monitoring flags.
<b>error</b>		Specifies debugging of health monitoring error flags.
<b>event</b>		Specifies debugging of health monitoring event flags.
<b>ha</b>		Specifies debugging of health monitoring HA flags.
<b>trace</b>		Specifies debugging of health monitoring trace flags.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug system** command is issued:

```
switch# debug system all
```

# debug tcap

To enable debugging the exception logger, use the **debug tcap** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug tcap demux | deque | error | info | init**

Syntax Description		
<b>demux</b>		Enables debugging for terminal capture demux functions.
<b>deque</b>		Enables debugging for terminal capture deque events.
<b>error</b>		Enables debugging for terminal capture errors.
<b>info</b>		Enables debugging for terminal capture information.
<b>init</b>		Enables debugging for terminal capture initialization.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** Use this command to debug terminal capture utility events and information.

**Examples** The following example displays the system output when the **debug demux** command is issued:

```
switch# debug demux
```

# debug tlport

To enable debugging for TL port interfaces, use the **debug tlport** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

## debug tlport

```
[ all (interface fc slot ) ] |
[ errors (interface fc slot ) ] |
[ events fc2 terminal (interface fc slot) | fc2 transit (interface fc slot) | mts (interface fc slot)
| pss (interface fc slot) ]
```

Syntax Description		
<b>all</b>		Enables debugging for all TL port features.
<b>errors</b>		Enables debugging for TL port error conditions.
<b>events</b>		Enables debugging for TL port monitoring events.
<b>fc2</b>		Enables debugging for TL port monitoring FC 2 events.
<b>terminal</b>		Specifies TL port monitoring FC 2 terminating events.
<b>transit</b>		Specifies TL port monitoring FC 2 transit events.
<b>mts</b>		Enables debugging for TL port monitoring MTS packets.
<b>pss</b>		Enables debugging for TL port monitoring PSS packets.
<b>interface fc slot</b>		Restricts debugging to the specified interface.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug tlport events pss** command is issued:

```
switch# debug tlport events pss
```

# debug ttyd

To enable TTYD debugging, use the **debug ttyd** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug ttyd all | errors | events**

<b>Syntax Description</b>	<b>all</b>	Enables debugging for all TTYD features.
	<b>errors</b>	Enables debugging for TTYD error conditions.
	<b>events</b>	Enables debugging for TTYD events.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug ttyd events** command is issued:

```
switch# debug ttyd events
```

# debug vni

To enable debugging for a virtual network interface (VNI), use the **debug vni** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug vni all | errors | events | info | pss**

Syntax Description		
<b>all</b>		Enables debugging for all VNI features.
<b>errors</b>		Enables debugging for VNI error conditions.
<b>events</b>		Enables debugging for VNI events.
<b>info</b>		Enables debugging for VNI events.
<b>pss</b>		Enables debugging for VNI PSS packets.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug vni all** command is issued:

```
switch# debug vni all
Apr 29 17:00:59 vni: Received MTS message
Apr 29 17:00:59 vni: message not processed by system mgr library , so process it normal
way
```

# debug vrrp

To enable debugging for a Virtual Router Redundancy Protocol (VRRP), use the **debug vrrp** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug vrrp
  [ configuration all | error | event | info ] |
  [ engine all | error | event | info ]
```

Syntax Description		
	<b>configuration</b>	Enables VRRP configuration debugging.
	<b>engine</b>	Enables VRRP engine debugging.
	<b>all</b>	Enables debugging for all VRRP features.
	<b>error</b>	Enables debugging for VRRP error conditions.
	<b>event</b>	Enables debugging for VRRP events.
	<b>info</b>	Enables debugging for VRRP events.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug vrrp engine all** command is issued:

```
switch# debug vrrp engine all
Apr 29 17:35:58 vrrp_eng: fu_priority_select: - setting fd[7] for select call - setting
fd[11] for select call - setting fd[12] for select call - setting fd [13] for select
call - setting fd[15] for select call
Apr 29 17:35:58 vrrp_eng: fu_priority_select_select_queue: round credit(6)
Apr 29 17:35:58 vrrp_eng: curr_q - FU_PSEL_Q_CAT_FD, usr_q_info(6), fd(15),
priority(2), credit(1), empty
Apr 29 17:35:58 vrrp_eng: fu_priority_select: returning FU_PSEL_Q_CAT_FD queue, fd(7),
usr_q_info(3)
Apr 29 17:35:58 vrrp_eng: heartbeat sent
Apr 29 17:35:58 vrrp_eng: message not processed by system mgr library , so process it
normal way
```

# debug vsan

To enable debugging for VSANs, use the **debug vsan** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug vsan [ all | global | ha | info | membership | mts ]**

Syntax Description		
<b>all</b>		Enables all debugging flags for the VSAN feature.
<b>global</b>		Enables debugging of events for the VSAN global parameter database
<b>ha</b>		Enables debugging of VSAN's HA-related events.
<b>info</b>		Enables debugging of events for VSAN information database.
<b>membership</b>		Enables debugging of events for VSAN membership database.
<b>mts</b>		Enables Tx/Rx packets of MTS.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug port all** command is issued:

```
switch# debug port-channel all
```



# debug wr-reg

To enable debugging for the list of devices using the write-register feature, use the **debug wr-reg** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug wr-reg** ( *device-name* | *register address* )

<b>Syntax Description</b>	<i>device-name</i>	Specifies the device name for the required device.
	<i>register address</i>	Specifies the register address for the required device.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug wr-reg** command is issued:

```
switch# debug wr-reg
```

# debug wwn

To enable debugging for the world wide name (WWN) manager, use the **debug wwn** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

**debug wwn all | detail | errors | flow | trace**

Syntax	Description
<b>all</b>	Enables all WWN debug options.
<b>detail</b>	Enables all WWN output
<b>error</b>	Enables debugging for WWN error conditions.
<b>flow</b>	Enables flow-level WWN debug options.
<b>trace</b>	Enables debugging for WWN traces.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug qos all** command is issued:

```
switch# debug wwn all
Apr 29 19:24:17 wwn: 53601-wwnm_sdwrap_dispatch:77|SDWRAP message Successfully processed
Apr 29 19:24:17 wwn: Src: 0x00000601/5206 Dst: 0x00000601/46 ID: 0x002C7DE4 Size: 252
[REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x002C7DE4 HA_SEQNO: 0x00000000 TS:
0x55D49A130243 REJ:0
Apr 29 19:24:17 wwn: 01 00 00 00 E7 03 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF
Apr 29 19:24:17 wwn: 2F 64 65 76 2F 70 74 73 2F 30 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr 29 19:24:17 wwn: 00 00 00 00 00 00 00 00 00 00 00 00 2E 00 00 00
Apr 29 19:24:17 wwn: 53601-wwnm_unmask_sigalarm:1261|TRACE:
FILE=_manager/wwnm/wwnm_utilities.c
```

# debug xbar

To enable crossbar debugging (XBAR), use the **debug xbar** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```

debug xbar
  [ all ] |
  [ demux ] |
  [ deque ] |
  [ error (module slot) ] |
  [ fsm (module slot) ] |
  [ ha (module slot) ] |
  [ init ] |
  [ main ]

```

Syntax Description		
<b>all</b>		Enables all XBAR debug options.
<b>demux</b>		Enables debugging for XBAR demux functions.
<b>deque</b>		Enables debugging for XBAR deque events.
<b>error</b>		Enables debugging for XBAR errors.
<b>fsm</b>		Enables debugging for XBAR FSMs.
<b>ha</b>		Enables debugging for XBAR high availability information.
<b>init</b>		Enables debugging for XBAR initialization.
<b>main</b>		Enables XBAR debugging for main functions.
<b>module slot</b>		Specifies the slot number of the module being debugged.

**Defaults** Enabled.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug xbar all** command is issued:

```

switch# debug xbar all
Apr 29 19:48:34 xbar: its a sdwrap msg, fsm utils dropping the mts msg
Apr 29 19:48:34 xbar: fu_fsm_engine: (Error) SYSERR_FU_xx: 0x10, err_num (16) in demux
Apr 29 19:48:34 xbar: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
Apr 29 19:48:34 xbar: fu_fsm_execute_all: null fsm_event_list
...

```

# debug xbc

To enable crossbar client debugging (XBC) , use the **debug xbc** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug xbc
  [ demux ] |
  [ deque ] |
  [ init ] |
  [ main ]
```

Syntax Description	Parameter	Description
	<b>demux</b>	Enables debugging for crossbar demux functions.
	<b>deque</b>	Enables debugging for crossbar deque events.
	<b>init</b>	Enables debugging for crossbar initialization.
	<b>main</b>	Enables debugging for crossbar main functions.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** Use this command to debug crossbar client events and information.

**Examples** The following example displays the system output when the **debug xbc init** command is issued:

```
switch# debug xbc init
```

# debug zone

To enable debugging for zones, use the **debug zone** command in EXEC mode. When you finish using a debug command, remember to disable it with its specific **no** command (or use the **no debug all** command to turn off all debugging).

```
debug zone
[ all ] |
[ change errors (vsan vsan-id) | events (vsan vsan-id) | packets (vsan vsan-id) ] |
[ database errors (vsan vsan-id) | events (vsan vsan-id) ] |
[ gs errors (vsan vsan-id) | events (vsan vsan-id) | packets (vsan vsan-id) ] |
[ merge errors (vsan vsan-id) | events (vsan vsan-id) | packets (vsan vsan-id) ] |
[ mts notifications (vsan vsan-id) ] |
[ pss errors (vsan vsan-id) | events (vsan vsan-id) ] |
[ tcam errors (vsan vsan-id) | events (vsan vsan-id) | packets (vsan vsan-id) ] |
```

Syntax	Description
<b>all</b>	Enables all zone server debug options.
<b>change</b>	Enables change protocol message debugging.
<b>errors</b>	Enables debugging for zone errors.
<b>events</b>	Enables debugging for zone events.
<b>packets</b>	Enables debugging   for zone packets.
<b>vsan vsan-id</b>	Restricts debugging to the specified VSAN.
<b>database</b>	Specifies database message debugging.
<b>gs</b>	Specifies GS protocol message debugging.
<b>merge</b>	Specifies merge protocol message debugging.
<b>mts notification</b>	Specifies MTS notification message debugging.
<b>tcam</b>	Specifies TCAM message debugging.

**Defaults** None.

**Command Modes** EXEC mode.

**Command History** This command was introduced in Cisco MDS SAN-OS Release 1.0(2).

**Usage Guidelines** None.

**Examples** The following example displays the system output when the **debug zone all** command is issued:

```
switch# debug zone all
```

